

**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO**

**CARLSON SANTANA CRUZ**

**HATEMILE: A BIBLIOTECA PARA GERAR PÁGINAS WEB  
MAIS ACESSÍVEIS**

**SÃO CRISTÓVÃO  
2015**

**UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO**

**CARLSON SANTANA CRUZ**

**HATEMILE: A BIBLIOTECA PARA GERAR PÁGINAS WEB  
MAIS ACESSÍVEIS**

Trabalho de Conclusão de Curso  
submetido ao Departamento de  
Computação da Universidade  
Federal de Sergipe como requisito  
parcial para a obtenção do título de  
Bacharel em Sistemas de  
Informação.

Orientador: Carlos Alberto Estombelo Montesco

**SÃO CRISTÓVÃO  
2015**

CRUZ, Carlson Santana.

HaTeMiLe: A biblioteca para gerar páginas web mais acessíveis / Carlson Santana Cruz – São Cristóvão: UFS, 2015.

149 folhas: fig., tab.

Trabalho de Conclusão de Curso – Universidade Federal de Sergipe, Curso de Sistemas de Informação, 2015.

1. Interação homem - computador. 2. Acessibilidade.  
3. Sistemas de Informação. I. HaTeMiLe: A biblioteca para gerar páginas web mais acessíveis.

**CARLSON SANTANA CRUZ**

**HATEMILE: A BIBLIOTECA PARA GERAR PÁGINAS WEB  
MAIS ACESSÍVEIS**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Computação da Universidade Federal de Sergipe (DCOMP/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

São Cristóvão, 10 de dezembro de 2015.

**BANCA EXAMINADORA:**

---

**Prof. Carlos Alberto Estombelo Montesco, Doutor  
Orientador  
DCOMP/UFS**

---

**Profa. Kenia Kodel Cox, Mestra  
DCOMP/UFS**

---

**Prof. Gilton José Ferreira da Silva, Mestre  
DCOMP/UFS**

CRUZ, Carlson Santana. **HaTeMiLe**: A biblioteca para gerar páginas web mais acessíveis. 2015. Trabalho de Conclusão de Curso – Curso de Sistemas de Informação, Departamento de Computação, Universidade Federal de Sergipe, São Cristóvão, 2015.

## RESUMO

*O número de usuários da internet aumenta a cada dia, assim como o número de serviços e informações disponibilizadas na web, para os deficientes visuais esses serviços representam a oportunidade de autonomia, porém frequentemente os deficientes visuais deparam-se com recursos na web que devido a sua deficiência ou tecnologia utilizada o impede ou dificulta seu uso, fazendo-os sentirem frustrados e limitados. Por isso o presente trabalho descreve o desenvolvimento de uma biblioteca e extensões para os navegadores Mozilla Firefox, Google Chrome e Opera e um plug-in para o WordPress que utilizam a biblioteca, capazes de tornarem páginas web mais acessíveis para os deficientes visuais, utilizando principalmente as recomendações do Web Content Accessibility Guidelines 2.0 e do Modelo de Acessibilidade do Governo Eletrônico 3.1 para prover soluções de acessibilidade, assim como os resultados dos testes que valida a eficácia da biblioteca.*

**Palavras-chave:** Acessibilidade na Web, Biblioteca, Automatiza, Deficiente, WCAG, e-MAG

## ABSTRACT

*The number of Internet users is increasing every day, as well as the number of services and informations available on the web, for visually impaired these services represent an opportunity to autonomy, but often the visually impaired are faced with resources on the web that due to their disability or technology used prevents or hinders their use, making their frustrated and limited. So this paper describes the development of a library and extensions for Mozilla Firefox, Google Chrome and Opera browsers and a plug-in for WordPress be able to make it accessible web pages, mainly based on the Web Content Accessibility Guidelines 2.0 and Modelo de Acessibilidade do Governo Eletrônico 3.1 guidelines to provide solutions of accessibility, as the results of tests that validate the effectiveness of library.*

**Key-words:** *Web Accessibility, Library, Automatize, disabled people, WCAG, e-MAG*

## **LISTA DE FIGURAS**

Figura 1- Diagrama de Classes do HaTeMiLe.....	21
Figura 2- Ícone do HaTeMiLe for Browser.....	43

## **LISTA DE TABELAS**

Tabela 1 - Relação entre os eventos via teclado e seus correspondentes via mouse.....	32
---	----



## LISTA DE ABREVIATURAS E SIGLAS

CMS	<i>Content Manager System</i>
CSS	<i>Cascading Style Sheet</i>
DOM	<i>Document Object Model</i>
e-MAG	Modelo de Acessibilidade do Governo Eletrônico
GPL	<i>GNU General Public License</i>
HPR	<i>Home Page Reader</i>
HTML	<i>HyperText Markup Language</i>
JAWS	<i>Job Access With Speech</i>
JSF	<i>JavaServer Faces</i>
NVDA	<i>NonVisual Desktop Access</i>
OO	Orientação a Objetos
PHP	<i>PHP Hypertext Preprocessor</i>
UFS	Universidade Federal de Sergipe
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>
WAVE	<i>Web Accessibility Evaluation Tool</i>
WCAG	<i>Web Content Accessibility Guidelines</i>
WebAIM	<i>Web Accessibility in Mind</i>
XML	<i>eXtensible Markup Language</i>

# SUMÁRIO

1	Introdução.....	12
1.1	Motivação.....	13
1.2	Objetivo Geral.....	14
1.3	Objetivos Específicos.....	14
1.4	Metodologia.....	15
2	Implementação da biblioteca.....	17
2.1	Linguagens de Programação.....	17
2.2	Arquivo de Configuração.....	18
2.3	Programação Orientada a Objetos.....	19
2.4	Código-fonte aberto.....	21
2.5	Apache License version 2.0.....	22
2.6	GitHub.....	22
2.7	Documentação.....	22
2.8	Controle de redundância, rastreabilidade e uniformidade.....	23
2.9	Visual da página.....	24
2.10	Limites, problemas e soluções.....	24
3	Soluções de acessibilidade da biblioteca.....	26
3.1	Documentos Técnicos.....	26
3.1.1	WCAG.....	26
3.1.2	e-MAG.....	29
3.2	Soluções implementadas.....	32
3.2.1	Disponibilizar todas as funções da página via teclado.....	32
3.2.2	Fornecer descrições para os campos.....	35
3.2.3	Fornecer instruções para a entrada de dados.....	35
3.2.4	Web semântica e WAI-ARIA.....	35
3.2.5	Fornecer âncoras para ir direto a um bloco de conteúdo.....	36
3.2.6	Associar células de dados às células de cabeçalho.....	36
3.2.7	Exibir a lista de atalhos da página.....	37
3.2.8	Prover uma forma de navegação para a descrição longa de um elemento não-textual.....	38
3.2.9	Promover a navegação entre os cabeçalhos da página.....	38
4	HaTeMiLe for Browser.....	40
4.1	Características de todas as extensões.....	40
4.1.1	Escolha das soluções executadas.....	40
4.1.2	Visual da página.....	41
4.1.3	Alterar textos.....	41
4.1.4	Internacionalização.....	41
4.1.5	Escopo das variáveis JavaScript.....	41
4.1.6	Ícone.....	42
4.1.7	Conflito de identificadores.....	43
4.1.8	jQuery.....	43
4.1.9	Frames.....	44
4.1.10	Alteração na biblioteca original.....	44
4.2	Extensão para o Mozilla Firefox.....	45
5	HaTeMiLe for WP.....	46

5.1 Extensão para o Google Chrome e a compatibilidade com o Opera.....	46
6 Testes realizados com o HaTeMiLe.....	48
6.1 Disponibilizar todas as funções da página via teclado.....	48
6.2 Fornecer descrições para os campos.....	49
6.3 Fornecer instruções para a entrada de dados.....	49
6.4 Web semântica e WAI-ARIA.....	50
6.5 Fornecer âncoras para ir direto a um bloco de conteúdo.....	50
6.6 Associar células de dados às células de cabeçalho.....	50
6.7 Exibir a lista de atalhos da página.....	50
6.8 Prover uma forma de navegação para a descrição longa de um elemento não-textual.....	51
6.9 Promover a navegação entre os cabeçalhos da página.....	51
7 Conclusão.....	52
Referências.....	53
APÊNDICE A – Saída em texto do NVDA.....	57
Página que testa todas as funcionalidades do HaTeMiLe, que não utiliza o HaTeMiLe.....	57
Página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe.....	77
Página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe que utiliza o CSS.....	104
Página que testa a semântica dos navegadores, que não utiliza o HaTeMiLe.....	131
Página que testa a semântica dos navegadores, que utiliza o HaTeMiLe.....	136
ANEXO I – E-mails da equipe de revisores das extensões da Mozilla.....	146
E-mail informando o motivo da recusa da versão 0.1.4 do HaTeMiLe for Browser.....	146
E-mail informando o motivo da versão 1.0.0 do HaTeMiLe for Browser não ter sido aprovada totalmente.....	147
E-mail informando o motivo da versão 1.0.0 do HaTeMiLe for Browser não ter sido aprovada totalmente, após ter sido informado os detalhes da versão.....	148

## 1 INTRODUÇÃO

A web é um meio de comunicação que está se tornando mais presente na vida das pessoas, os sites gradativamente possuem designs mais elaborados e mais recursos que aumentam a satisfação, facilidade, informação e funcionalidade para os usuários, porém as pessoas com deficiência necessitam que os sites sejam acessíveis para poder utilizá-los (TAKAGI et al., 2007). Os deficientes podem utilizar tecnologias que os auxiliam a utilizar os recursos do computador, as tecnologias assistivas, essas tecnologias assim como a acessibilidade na web podem ajudar outros tipos de usuários a acessarem melhor um site, entre esses usuários estão os idosos, as pessoas com deficiência temporária decorrido de acidente ou doença, entre outros (CHALEGRE, 2011; ROCHA; DUARTE, 2013a).

Por isso é importante entender o funcionamento dos agentes de usuários e das tecnologias assistivas utilizadas pelos usuários para compreender suas limitações e melhorar a acessibilidade dos sites, mas também é necessário perceber outros fatores mais subjetivos para melhorar a acessibilidade dos sites (ROCHA; DUARTE, 2013b).

No intuito de promover uma web acessível o *World Wide Web Consortium* (W3C) criou a *Web Accessibility Initiative* (WAI), que produz documentos com recomendações de acessibilidade para que os desenvolvedores de sites possam projetar sites com a finalidade de atender uma gama maior de usuários, como a *Web Content Accessibility Guidelines* (WCAG). Além disso há muitas ferramentas disponíveis na internet para a validação da acessibilidade na web a maioria baseado nas recomendações do WCAG 2.0 ou WCAG 1.0, como o *Web Accessibility Evaluation Tool* (WAVE) da *Web Accessibility in Mind* (WebAIM), o DaSilva, *Achecker*, entre outros, que apesar de serem importantes ferramentas para a validação da acessibilidade de sites, os mesmos possuem limitações que faz com que seja necessário ainda seja necessário pessoas também realizarem testes (CHALEGRE, 2011; CENTENO et al., 2006).

Também no intuito de promover a acessibilidade, muitos países como os Estados Unidos da América, Canadá, Portugal, Espanha, Inglaterra, entre outros, possuem legislações que dispõe sobre a acessibilidade na web e até mesmo a Organização das Nações Unidas garante como um direito humano o acesso a web sem nenhuma forma de barreira. No Brasil o

decreto nº 5.296 de 2 de dezembro de 2004 determina que os sites mantidos por órgãos do governo federal sejam obrigados a ser acessíveis, utilizando as recomendações do Modelo de Acessibilidade do Governo Eletrônico (e-MAG), infelizmente a iniciativa privada esta isenta dessa obrigação (LOIACONO; DJAMASBI; KIRYAZOV, 2013; ROCHA; DUARTE, 2013a).

Baseados no WCAG, muitos países criaram seus próprios documentos e normas sobre acessibilidade na web, inclusive o Brasil para formular o e-MAG. No caso do e-MAG 3.1 quase todas suas recomendações, possuem recomendações equivalentes as recomendações do WCAG 2.0, com poucas recomendações que estão presentes num documento que não possui uma recomendação equivalente no outro, mas as recomendações básicas que garantem o acesso estão disponíveis em ambos os documentos. Por isso ao se garantir que se cumpriu todas as recomendações num documento, pode-se garantir que tenha se cumprido a maioria das recomendações no outro documento (ROCHA; DUARTE, 2012).

O presente trabalho propõe a criação de uma biblioteca capaz de automatizar parte da tarefa de tornar um site acessível convertendo o seu código *HyperText Markup Language* (HTML) em outro mais acessível.

## **1.1 Motivação**

Apesar do número de pessoas com deficiência crescer a cada ano, principalmente devido ao aumento da perspectiva de vida população, 98% das páginas HTML não são acessíveis (CHALEGRE, 2011). Essas pessoas com deficiência muitas vezes vem a internet como um meio de acesso onde elas podem obter informações e realizar operações sem necessitar do auxílio de outras pessoas, o que acaba frustrando-as quando se deparam com a inacessibilidade nos sites por fazer se sentirem limitadas e dependentes. (ROCHA; DUARTE, 2013a; TAKAGI et al., 2007).

E apesar de muitas iniciativas serem criadas na tentativa de dar o acesso para as pessoas com deficiência, ao se deparar com projetos de construção de sites os programadores acabam se preocupando mais com as funcionalidades, design, usuário, eficiência, tecnologias, etc; e acabam deixando de lado a acessibilidade ou mesmo desconhece métodos para melhorá-la (GUERCIO et al., 2011). Por isso se faz necessário o desenvolvimento de tecnologias que auxiliem os desenvolvedores a manterem sites mais acessíveis e também tecnologias que

façam com que os usuários com deficiência não dependam dos desenvolvedores para terem acessibilidade, no caso desse trabalho a biblioteca desenvolvida pretende ser utilizada tanto pelos desenvolvedores quanto pelos usuários finais para melhorar a acessibilidade na web.

## 1.2 Objetivo Geral

Desenvolver uma biblioteca capaz de melhorar a acessibilidade das páginas web para as pessoas com deficiência, principalmente visual.

## 1.3 Objetivos Específicos

Visando alcançar o objetivo principal, alguns objetivos específicos são requeridos, entre eles:

- Pesquisar quais são os problemas de acessibilidade que os deficientes possuem ao navegar na web;
- Implementar a biblioteca de forma que a mesma seja capaz de diminuir ou acabar com os problemas de acessibilidade, que não sejam referentes a sintaxe ou semântica, encontrados no código HTML, gerando um novo código HTML que evita causar novos problemas de acessibilidade;
- Implementar a biblioteca de modo que o código HTML gerado pela biblioteca, mesmo adicionando novos atributos a elementos ou novos elementos no código HTML, possa manter o visual original da página, podendo ser utilizado *Cascading Style Sheet* (CSS) para isso;
- Implementar a biblioteca de modo que o código HTML gerado pela biblioteca apenas disponibilize as informações já contidas na página e não modifique nenhuma regra de negócio;
- Implementar a biblioteca de modo que o código HTML gerado por ela seja compatível com o HTML5 e o HTML 4.01, devendo ser compatível com o WebVox quando não incluir JavaScript, e só deverá incluir JavaScript no código HTML gerado quando o que se deseja corrigir é algo relacionado com JavaScript, sendo que esse JavaScript deverá ser compatível com o Mozilla Firefox 4.0 e o Internet Explorer 8;

- Permitir que os textos inseridos pela biblioteca possam ser definidos pelo desenvolvedor ou usuário;
- Desenvolver a biblioteca nas linguagens de programação mais utilizadas na web, para garantir que a biblioteca possa ser utilizada por desenvolvedores, no *server-side*, aplicações que são executadas pelo servidor, e no *client-side*, aplicações que são executadas pelo cliente, devendo todas as funcionalidades contidas em uma versão da biblioteca implementada em um linguagem de programação disponíveis em todas as outras;
- Permitir que a biblioteca possa ser utilizada, estendida e modificada por qualquer pessoa, inclusive de outros países;
- Implementar a biblioteca de forma flexível, para que possa ser utilizada em conjunto com diversos softwares.

#### 1.4 Metodologia

A metodologia do projeto se dará da seguinte forma:

1. Coleta de informações de quais são os problema de acessibilidade enfrentados pelas pessoas com deficiência ao navegar pela Internet, através dos documentos da W3C, documentos utilizados pelo Brasil e outros países como normatizadores, artigos científicos, artigos de sites conceituados e manuais de softwares;
2. Verificar quais problemas podem ser resolvidos de forma automatizada;
3. Implementar a solução na biblioteca;
4. Desenvolver um software que utilize a biblioteca para poder ser utilizado pelos deficientes:
  1. Esse software deverá ser a extensão dos navegadores utilizados pelos deficientes visuais;
  2. Os usuários do software poderão decidir quais são as funcionalidades que serão executadas pelo software;
  3. Os usuários do software poderão alterar os textos gerados pela biblioteca.
5. Verificar e validar a biblioteca para saber se houve melhoras sentidas na acessibilidade

das páginas

6. Avaliar os resultados obtidos nas validações.



## 2 IMPLEMENTAÇÃO DA BIBLIOTECA

HaTeMiLe é o nome da biblioteca de código-fonte aberto desenvolvida neste trabalho que procura melhorar a acessibilidade de um código HTML transformando em um novo código HTML mais acessível. O nome HaTeMiLe foi escolhido por ser um anagrama para *HTML Accessible*, removendo as consoantes de *Accessible*.

A ideia surgiu devido a observação da falta de acessibilidade dos *frameworks* de componentes do *JavaServer Faces* (JSF), principalmente o PrimeFaces 4.0 que possuem componentes que poderiam ser utilizados via teclado, porém só podem utilizados via mouse, com isso a ideia inicial era desenvolver extensões para *frameworks* web no *server-side* de diversas linguagens de programação e uma biblioteca no *client-side*, porém foi percebido que existe um número muito grande de *frameworks* web e que os usuários finais poderiam se interessar em utilizar os recursos da biblioteca mesmo em sites que não utilizassem o *framework* ou a biblioteca. Devido a isso o HaTeMiLe foi desenvolvido em mais de uma linguagem de programação para permitir que possa ser implementado em conjunto com outros softwares tanto no *client-side* como no *server-side*, esses softwares podem ser agentes de usuário, *frameworks* web, servidores web, editores de texto, entre outros.

### 2.1 Linguagens de Programação

A biblioteca foi implementada nas linguagens de programação: JavaScript e CoffeeScript, disponível em <<https://github.com/carlsonsantana/HaTeMiLe-for-JavaScript>>; *PHP Hypertext Preprocessor* (PHP), disponível em <<https://github.com/carlsonsantana/HaTeMiLe-for-PHP>>; Java, disponível em <<https://github.com/carlsonsantana/HaTeMiLe-for-Java>>; Ruby, disponível em <<https://github.com/carlsonsantana/HaTeMiLe-for-Ruby>>; Python, disponível em <<https://github.com/carlsonsantana/HaTeMiLe-for-Python>>. Abaixo é listado os motivos da escolha de cada linguagem de programação:

- A linguagem JavaScript foi escolhida por ser a linguagem de programação padrão utilizada para manipular o comportamento dos documentos HTML, sendo utilizada no

*client-side* (W3C, 2014a);

- A linguagem CoffeeScript foi escolhida por ser simples e o mais importante, para traduzir os códigos escritos em CoffeeScript para códigos JavaScript eficientes e compatíveis com diversos navegadores (COFFEESCRIPT, [2014?]);
- A linguagem PHP foi escolhida por ser a linguagem de programação mais utilizada para se desenvolver sites no *server-side* atualmente (W3TECHS, 2014);
- A linguagem Java foi escolhida por ser a linguagem de programação mais utilizada (CASS, 2014), e também por seu *framework* JSF, utilizado em aplicações web, ser inacessível (AKQUINET, 2013);
- Ruby e Python foram escolhidas por serem as linguagens de programação mais utilizadas entre os usuários do GitHub e StackExchange (FINLEY, 2012).

Os nomes dos parâmetros, classes e métodos públicos são iguais em todas as linguagens de programação, sendo escritos em inglês, para facilitar o entendimento das funcionalidades por desenvolvedores de outros países.

## 2.2 Arquivo de Configuração

A biblioteca utiliza o *eXtensible Markup Language* (XML) como arquivo de configuração para as versões *server-side*, esse formato foi escolhido porque todas as linguagens de programação escolhidas possuem suporte nativo. Para JavaScript entretanto a configuração é realizada através da passagem de um objeto, essa diferença existe porque um arquivo XML com os mesmos nomes de parâmetros é mais pesado do que uma configuração através de um objeto JavaScript.

Apesar da diferença entre as linguagens de programação e os formatos, os nomes dos parâmetros são iguais em todos eles, assim como os nomes dos métodos os nomes dos parâmetros estão em inglês, isso para que a biblioteca possa ser utilizada por pessoas de outros países.

## 2.3 Programação Orientada a Objetos

A biblioteca foi implementada utilizando o paradigma da Orientação a Objetos (OO), esse paradigma foi escolhido porque todas as linguagens de programação escolhidas possuem suporte a esse paradigma, a manutenção do código-fonte se torna mais simples e isso permite que classes e métodos possam ser reaproveitados, o que é desejável em uma biblioteca de código-fonte aberto, pois isso permite que outros desenvolvedores possam dar desenvolver novos recursos e solucionar erros de implementação (HOCK-CHUAN, 2013).

Todas as versões das bibliotecas implementadas utilizam algum recurso para gerar e manipular o *Document Object Model* (DOM) do código HTML passado e utilizar recursos como seletores CSS para encontrar elementos no código HTML, como implementar esse tipo de solução não é o objetivo final da biblioteca se optou por utilizar bibliotecas de terceiros para facilitar a implementação do HaTeMiLe, sendo utilizado as bibliotecas jQuery na versão em JavaScript, Jsoup na versão em Java, phpQuery na versão em PHP, Nokogiri na versão em Ruby e BeautifulSoup na versão em Python. E como cada biblioteca escolhida em cada linguagem de programação possui sua própria forma de manipular o DOM, foi necessário definir interfaces para a manipulação do elemento especificamente a interface `hatemile.util.HTMLDOMEElement` e para a manipulação do parser a interface `hatemile.util.HTMLDOMParser`, essas interfaces permitem também que possam ser utilizadas novas bibliotecas para a manipulação do código HTML sem ser necessário modificar outras classes da biblioteca.

Também foram implementadas interfaces para as soluções da biblioteca isso para permitir que os desenvolvedores possam implementar novas soluções para os problemas de acessibilidade que a biblioteca procura resolver:

- A interface `hatemile.AccessibleEvent` é responsável pelas soluções referentes aos eventos;
- A interface `hatemile.AccessibleForm` é responsável pelas soluções referentes aos formulários;
- A interface `hatemile.AccessibleImage` é responsável pelas soluções referentes a imagens;

- A interface `hatemile.AccessibleNavigation` é responsável pelas soluções referentes a navegação da página;
- A interface `hatemile.AccessibleSelector` é responsável pelas soluções que utilizam seletores CSS;
- A interface `hatemile.AccessibleTable` é responsável pelas soluções referentes a tabelas.

Apenas nas versões *client-side* da biblioteca as classes não utilizam as Interfaces, pois para isso seria necessário incluir essas interfaces aos *scripts* da página o que aumenta a banda necessária para baixar a página.

As outras classes são auxiliares a biblioteca:

- A classe `hatemile.util.Configure` é responsável pela configuração do HaTeMiLe;
- A classe `hatemile.util.CommonFunctions` é responsável por fornecer funções que auxiliam no desenvolvimento da biblioteca;
- A classe `hatemile.util.SelectorChange` é responsável por armazenar as mudanças que serão aplicadas nos elementos encontrados pelos seletores CSS;
- A classe `hatemile.util.Skipper` é responsável por armazenar os textos e os atalhos dos links que serão criados a partir dos elementos encontrados pelos seletores CSS.

Na Figura 1 é mostrado o diagrama de classes da biblioteca sem as implementações das classes que utilizam as bibliotecas que manipulam o código HTML e classes que implementam a soluções da biblioteca:

O código-fonte da biblioteca é aberto para permitir que qualquer desenvolvedor possa estudar, modificar e distribuir a biblioteca modificada ou softwares que utilizem a biblioteca, mesmo que seja para fins comerciais ou em softwares proprietários. Com isso se espera que outros desenvolvedores possam oferecer novas soluções para os problemas que a biblioteca

atualmente tenta resolver, resolver outros problemas de acessibilidade ou dar suporte a novos recursos, como outros *parsers* por exemplo.

Para distribuir o código-fonte da biblioteca foi escolhida a licença *Apache License version 2.0* e para compartilhar o código-fonte foi escolhido GitHub.

## 2.5 Apache License version 2.0

A licença *Apache License version 2.0* foi escolhida porque permite que o código-fonte da biblioteca possa ser utilizada em softwares de código-fonte aberto ou proprietários, além de permitir o uso, a venda desses softwares e a sua compatibilidade com outras licenças, como a licença *GNU General Public License Version 3 (GPLv3)*, *MIT License*, *BSD License*, *Creative Commons*, entre outras licenças para softwares livres e de código-fonte aberto (APACHE, [2007?], [2004?]).

## 2.6 GitHub

O GitHub foi escolhido para hospedar o código-fonte da biblioteca, pois ele fornece repositórios gratuitos para projetos de código-fonte aberto utilizando os sistemas de controle de versão Git e SubVersion (GITHUB, [20--]), sendo possível localizar os projetos através do mecanismo de busca interno ou mesmo através de mecanismos de buscas externo como o Google e o Bing.

Entre os recursos disponíveis para o GitHub estão a segurança, sistema para gerenciamento de tarefas, contato com outros desenvolvedores, criação de projetos baseados em outros já existentes no GitHub, visualização e edição do código-fonte com formatação, controle de submissão de mudanças, visualizador de diferenças, etc (GITHUB, [20--]).

## 2.7 Documentação

O código-fonte da biblioteca é aberto, mas para que o seu código-fonte possa ser melhor

compreendido por outros desenvolvedores é necessário que ele possua uma documentação explicando a finalidade de suas classes, métodos e parâmetros. A documentação foi escrita em inglês para que o código-fonte da biblioteca possa ser compreendido por desenvolvedores de outros países.

Assim como as classes e métodos da biblioteca possuem os mesmos nomes, parâmetros, objetivos e soluções nas suas versões de cada linguagem de programação, a documentação utiliza os mesmos textos para descrever suas finalidades, podendo diferenciar apenas nas classes e métodos privados por poderem cuidar de problemas específicos da linguagem de programação ou de uma biblioteca utilizada. Mas apesar dos textos serem iguais, há diversas formas e ferramentas para documentar o código-fonte.

Abaixo é listado quais foram os softwares utilizados para documentar o código-fonte das versões da biblioteca em cada linguagem de programação:

- O jsDoc foi escolhido por ser a ferramenta de documentação parecida com o JavaDoc para o JavaScript, além de possuir recursos para tornar a documentação mais rica. Mesmo o CoffeeScript possuindo ferramentas de documentação própria, o jsDoc foi mantido para que o código-fonte do CoffeeScript ao ser convertido para JavaScript mantenha a documentação;
- O JavaDoc foi escolhido por ser a ferramenta de documentação padrão para o Java, além disso possui uma série de recursos que tornam a documentação mais rica;
- O phpDocumentor foi escolhido por ser uma ferramenta de documentação parecida com o JavaDoc para o PHP, além disso possui uma série de recursos que tornam a documentação mais rica;
- O Epydoc foi escolhido, pois apesar de não ser a ferramenta de documentação padrão, as DocStrings, para o Python, possui compatibilidade com as DocStrings, além de disso possui uma série de recursos que tornam a documentação mais rica;
- O RDoc foi escolhido por ser a ferramenta de documentação padrão do Ruby.

## **2.8 Controle de redundância, rastreabilidade e uniformidade**

A redundância é um fator que pode atrapalhar a usabilidade da página, por isso a biblioteca

evita causar problemas de redundância na página. Para evitar a redundância cada novo elemento adicionado pela biblioteca possui um atributo rastreável para que seja possível identificar que determinada solução de acessibilidade foi executada, assim ao ser executada a mesma solução para o mesmo elemento será verificado se existe algum elemento com um atributo rastreável que referencia o elemento, assim havendo algum atributo rastreável de uma solução que referencia o elemento, o elemento que seria adicionado ao código HTML não é mais adicionado na página.

Devido a biblioteca utilizar a mesma solução para resolver os problemas em todas as suas versões e utilizar os mesmos atributos rastreáveis para todas as versões da biblioteca, esse recurso também evita a redundância quando a biblioteca é executada no *server-side* e no *client-side*, entre outros casos que possam acontecer.

## **2.9 Visual da página**

A intenção da biblioteca é que mesmo sendo utilizada em sites com um visual que não deva ser alterado, a biblioteca possa ser utilizada.

Por isso cada elemento adicionado no novo código HTML possui um identificador único ou uma classe, isso faz com que seja possível ocultar esses novos elementos e manter o visual da página através do CSS. Porém devem ser tomados cuidados ao ocultar o elemento, isso porque a depender da forma como é feito, o elemento pode ser ocultado também do leitor de tela tornando a utilização de certas soluções inúteis.

O e-MAG 3.1 aconselha o uso de determinadas propriedades CSS para se ocultar apenas visualmente o elemento, essas propriedades podem ser utilizadas para ocultar os elementos adicionados pela biblioteca. Há também a possibilidade de se utilizar o CSS para adaptar os elementos adicionados pela biblioteca ao visual da página ao invés de ocultá-los, o que torna a página ainda mais acessível, pois os seus recursos estarão disponíveis para todos os usuários ao invés de apenas para os usuários que utilizam leitores de tela.

## **2.10 Limites, problemas e soluções**



A biblioteca possui um escopo definido do que não deve corrigir: os problemas sintáticos, semânticos, visuais, de navegação, de comportamento e de contexto do código HTML. E apesar dessas categorias de problemas possuírem mais problemas de acessibilidade, a biblioteca não tem o objetivo de resolvê-los, isso porque a ideia inicial da biblioteca era ser utilizada apenas por desenvolvedores através de *frameworks* web e o que se deseja é que mesmo sendo utilizada a biblioteca, o seu uso não cause vícios ao desenvolvedor ou aos produtores de conteúdo.

Da mesma forma que o desenvolvedor pode querer que a biblioteca tente resolver determinado problema de acessibilidade para algum elemento da página, ele pode entender que determinado elemento não deve ser manipulado, por isso a biblioteca utiliza um atributo no elemento que informa para a biblioteca para não resolver determinado problema de acessibilidade para o elemento.

### 3 SOLUÇÕES DE ACESSIBILIDADE DA BIBLIOTECA

O HaTeMiLe modifica o código HTML que foi passado para a biblioteca, procurando retornar um código HTML mais acessível para o software que o chamou. Para tornar o código HTML mais acessível foram utilizadas como base os documentos WCAG 2.0 e o e-MAG 3.1, algumas funcionalidades do *Job Access With Speech (JAWS)*, versões do Opera anteriores a 15 e Mozilla Firefox. Os navegadores utilizados para realizar os testes de compatibilidade com suporte a JavaScript foram o Mozilla Firefox 4.0, Mozilla Firefox 40.0, Google Chrome 39.0.2171.71, Opera 12.16.1860, Opera 31.0, Safari 5.34.57.2 e Internet Explorer 9, para testes de compatibilidade sem suporte a JavaScript foram utilizados os navegadores Lynx 2.8.8pre.4 e WebVox na versão 5.5 do DOSVOX.

#### 3.1 Documentos Técnicos

A biblioteca utilizou como base de conhecimento principal para conhecer os problemas de acessibilidade o WCAG 2.0 e o e-MAG 3.1.

##### 3.1.1 WCAG

Em 1999 o W3C criou a WAI, cuja uma de suas funções é elaborar recomendações para que a Web seja acessível para as pessoas com deficiências ou em condições especiais de acesso, esse grupo publicou no mesmo ano a primeira versão da WCAG (WCAG 1.0), um documento com catorze recomendações de acessibilidade para sites. Em 2008, o WCAG foi atualizado para a versão 2.0, embora não tenham substituído ou invalidado a primeira versão, têm seu uso recomendado para a criação de novos conteúdos e para a atualização dos já existentes (ROCHA; DUARTE, 2012). O WCAG 2.0 também foi aprovado como um padrão ISO, no caso a ISO/IEC 40500:2012 (W3C, 2012).

Sendo que apenas as seguintes recomendações foram corrigidas pela biblioteca:

- 1.3.1 - *Info and Relationships*;

- 1.3.3 - *Sensory Characteristics*;
- 2.1.1 - *Keyboard*;
- 2.1.3 - *Keyboard (No Exception)*;
- 2.4.1 - *Bypass Blocks*;
- 4.1.2 - *Name, Role, Value*.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Contexto**:

- 1.1.1 - *Non-text Content*;
- 1.2.1 - *Audio-only and Video-only (Prerecorded)*;
- 1.2.2 - *Captions (Prerecorded)*;
- 1.2.3 - *Audio Description or Media Alternative (Prerecorded)*;
- 1.2.4 - *Captions (Live)*;
- 1.2.5 - *Audio Description (Prerecorded)*;
- 1.2.6 - *Sign Language (Prerecorded)*;
- 1.2.7 - *Extended Audio Description (Prerecorded)*;
- 1.2.8 - *Media Alternative (Prerecorded)*;
- 1.2.9 - *Audio-only (Live)*;
- 2.4.2 - *Page Titled*;
- 2.4.3 - *Focus Order*;
- 2.4.4 - *Link Purpose (In Context)*;
- 2.4.6 - *Headings and Labels*, porém a biblioteca utiliza essa recomendação para outras funcionalidades;
- 2.4.9 - *Link Purpose (Link Only)*;
- 2.4.10 - *Section Headings*, porém a biblioteca utiliza essa recomendação para outras funcionalidades;
- 3.1.1 - *Language of Page*;
- 3.1.2 - *Language of Parts*;
- 3.1.3 - *Unusual Words*;
- 3.1.4 - *Abbreviations*;

- 3.1.5 - *Reading Level*;
- 3.1.6 - *Pronunciation*;
- 3.3.2 - *Labels or Instructions*, porém a biblioteca utiliza essa recomendação para outras funcionalidades;
- 3.3.3 - *Error Suggestion*;
- 3.3.5 - *Help*;
- 3.3.6 - *Error Prevention (All)*.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Visual**:

- 1.3.2 - *Meaningful Sequence*;
- 1.4.1 - *Use of Color*;
- 1.4.2 - *Audio Control*;
- 1.4.3 - *Contrast (Minimum)*;
- 1.4.4 - *Resize text*;
- 1.4.5 - *Images of Text*;
- 1.4.6 - *Contrast (Enhanced)*;
- 1.4.7 - *Low or No Background Audio*;
- 1.4.8 - *Visual Presentation*;
- 1.4.9 - *Images of Text (No Exception)*;
- 2.3.1 - *Three Flashes or Below Threshold*;
- 2.3.2 - *Three Flashes*;
- 2.4.7 - *Focus Visible*.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Comportamento da página**:

- 2.2.1 - *Timing Adjustable*;
- 2.2.2 - *Pause, Stop, Hide*;
- 2.2.3 - *No Timing*;
- 2.2.4 - *Interruptions*;
- 2.2.5 - *Re-authenticating*;

- 3.2.1 - *On Focus*;
- 3.2.2 - *On Input*;
- 3.2.5 - *Change on Request*;
- 3.3.1 - *Error Identification*.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Navegação**:

- 2.4.5 - *Multiple Ways*;
- 2.4.8 - *Location*;
- 3.2.3 - *Consistent Navigation*;
- 3.2.4 - *Consistent Identification*.

As recomendações **2.1.2 - No Keyboard Trap** e **4.1.1 - Parsing** não foram resolvidas, pois se tratam de problemas que **Não envolve HTML e Sintaxe e Semântica**, respectivamente.

### 3.1.2 e-MAG

O e-MAG foi desenvolvido em 2004 baseado no estudo de 14 normas existentes em outros países acerca de acessibilidade digital. Sendo que suas versões 1.4 e 2.0 eram divididas em dois documentos, a visão do cidadão e a cartilha técnica. Essa divisão do e-MAG apresentou alguns inconvenientes durante o processo de disseminação do modelo, como a dificuldade das pessoas entenderem as áreas da Visão do Cidadão e seu relacionamento com a aplicação efetiva da acessibilidade. O aprendizado durante os seis anos da versão 2.0 do e-MAG e o lançamento da versão 2.0 do WCAG em 2008 marcaram o caminho para a revisão do modelo que resultou na versão 3.0 em 21 de setembro de 2011. Apesar de utilizar a WCAG como referência, e estar alinhado a esta, o e-MAG 3.0 foi desenvolvido e pensado para as necessidades locais, visando atender as prioridades brasileiras, essa versão unificou em apenas um documento o modelo, retirando a separação entre visão técnica e visão do cidadão, também foi abandonado os níveis de prioridade A, AA e AAA, visto que o padrão é voltado as páginas do Governo, não sendo permitido exceções com relação ao cumprimento das recomendações. Além disso, no e-MAG 3.0 foi incluída a seção chamada “Padronização de

acessibilidade nas páginas do governo federal”, com o intuito de padronizar elementos de acessibilidade que devem existir em todos os sítios e portais do governo (BRASIL, 2014).

Sendo que apenas as seguintes recomendações foram corrigidas pela biblioteca:

- 1.5 - Fornecer âncoras para ir direto a um bloco de conteúdo;
- 1.9 - Não abrir novas instâncias sem a solicitação do usuário;
- 2.1 - Disponibilizar todas as funções da página via teclado;
- 3.7 - Utilizar mapas de imagem de forma acessível;
- 3.10 - Associar células de dados às células de cabeçalho;
- 6.2 - Associar etiquetas aos seus campos;
- 6.5 - Fornecer instruções para entrada de dados.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Contexto**:

- 1.3 - Utilizar corretamente os níveis de cabeçalho;
- 1.4 - Ordenar de forma lógica e intuitiva a leitura e tabulação;
- 1.6 - Não utilizar tabelas para diagramação;
- 1.7 - Separar links adjacentes;
- 1.8 - Dividir as áreas de informação;
- 2.2 - Garantir que os objetos programáveis sejam acessíveis;
- 2.7 - Assegurar o controle do usuário sobre as alterações temporais do conteúdo;
- 3.1 - Identificar o idioma principal da página;
- 3.2 - Informar mudança de idioma no conteúdo;
- 3.3 - Oferecer um título descritivo e informativo à página;
- 3.5 - Descrever links clara e sucintamente;
- 3.6 - Fornecer alternativa em texto para as imagens do sítio;
- 3.9 - Em tabelas, utilizar títulos e resumos de forma apropriada;
- 3.11 - Garantir a leitura e compreensão das informações;
- 3.12 - Disponibilizar uma explicação para siglas, abreviaturas e palavras incomuns;
- 4.2 - Não utilizar apenas cor ou outras características sensoriais para diferenciar elementos;

- 5.1 - Fornecer alternativa para vídeo;
- 5.2 - Fornecer alternativa para áudio;
- 5.3 - Oferecer audiodescrição para vídeo pré-gravado;
- 5.4 - Fornecer controle de áudio para som;
- 6.1 - Fornecer alternativa em texto para os botões de imagem de formulários;
- 6.3 - Estabelecer uma ordem lógica de navegação;
- 6.7 - Agrupar campos de formulário.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Visual**:

- 2.6 - Não incluir situações com intermitência de tela;
- 4.1 - Oferecer contraste mínimo entre plano de fundo e primeiro plano;
- 4.3 - Permitir redimensionamento sem perda de funcionalidade;
- 4.4 - Possibilitar que o elemento com foco seja visualmente evidente;
- 5.5 - Fornecer controle de animação.

As seguintes recomendações não foram resolvidas pela biblioteca por ser tratar de problema de **Comportamento da página**:

- 2.3 - Não criar páginas com atualização automática periódica;
- 2.4 - Não utilizar redirecionamento automático de páginas;
- 2.5 - Fornecer alternativa para modificar limite de tempo;
- 6.4 - Não provocar automaticamente alteração no contexto;
- 6.6 - Identificar e descrever erros de entrada de dados e confirmar o envio das informações;
- 6.8 - Fornecer estratégias de segurança específicas ao invés de CAPTCHA.

As recomendações **1.1 - Respeitar os Padrões Web** e **1.2 - Organizar o código HTML de forma lógica e semântica**, **3.4 - Informar o usuário sobre sua localização na página** e **3.8 - Disponibilizar documentos em formatos acessíveis** não foram resolvidas, pois se tratam de problemas que **Sintaxe e Semântica**, **Navegação** e **Não envolve HTML** respectivamente.

## 3.2 Soluções implementadas

### 3.2.1 Disponibilizar todas as funções da página via teclado

Este problema é citado pela recomendação 2.1 do e-MAG 3.1 e pelas recomendações 2.1.1 e 2.1.3 do WCAG 2.0, para automatizar essa solução é necessário saber o que faz com que a navegação seja inacessível via teclado, o e-MAG 3.1 e o *Techniques for WCAG 2.0* cita eventos JavaScript específicos para o mouse, informando inclusive seus eventos equivalentes via teclado.

Tabela 1 - Relação entre os eventos via teclado e seus correspondentes via mouse

Evento do teclado	Evento correspondente do mouse
keydown	mousedown
keyup	mouseup
keypress	click
focus	mouseover
blur	mouseout

Porém tanto o e-MAG 3.1 quanto o *Techniques for WCAG 2.0* não contemplam os eventos relacionados a *Drag-and-Drop* (Arraste e Solte) que é um recurso nativo do HTML5, que durante a pesquisa foi percebido que o recurso não possui suporte nativo pelo teclado nos navegadores Mozilla Firefox, Google Chrome, Opera, Internet Explorer e Safari, necessitando que os desenvolvedores implementem formas de acesso alternativas para o teclado, os navegadores WebVox e Lynx não foram testados, pois os mesmos não possuem suporte a JavaScript que é necessário para utilizar o *Drag-and-Drop* do HTML5.

Para procurar os elementos que possuem eventos inacessíveis via teclado, é necessário saber como eles são associados aos elementos e como recuperar esses eventos. Para associar e recuperar um evento de um elemento, é possível utilizar os atributos dos elementos ou *scripts*, porém apesar dos eventos associados via atributos poderem ser recuperados via *scripts*, o inverso não é possível.



Na versão da biblioteca que executa no *server-side* os elementos que possuem eventos inacessíveis via teclado são encontrados apenas quando eles possuem os respectivos atributos nos elementos: *onmousedown*, *onmouseup*, *onclick*, *ondblclick*, *onmouseover*, *onmouseout*, *ondrag*, *ondragstart*, *ondragend*, *ondragenter*, *ondragover*, *ondragleave* ou *ondrop*. Já na versão da biblioteca que executa no *client-side*, apesar de ser possível localizar os elementos pelos seus atributos, os elementos são encontrados quando possuem os seguintes métodos nos objetos do DOM ou na lista de funções do evento do elemento: *mousedown*, *mouseup*, *click*, *dblclick*, *mouseover*, *mouseout*, *drag*, *dragstart*, *dragend*, *dragenter*, *dragover*, *dragleave* ou *drop*. Mas nativamente não é possível saber se nessa lista há algum evento, por isso é necessário incluir o *script* que modifica a forma como as funções são adicionadas ou removidas da lista, devendo ser o primeiro *script* a ser executado na página, por isso o próprio desenvolvedor deve incluí-lo no lado do cliente. Essa diferença na forma como os elementos são encontrados faz com que a versão *client-side* da biblioteca seja mais eficaz que a versão *server-side*.

Após localizar os elementos é necessário: disponibilizar o acesso ao elemento via teclado, caso o mesmo não possua; incluir o *script* que modifica a forma como as funções são adicionadas ou removidas na lista de funções dos eventos do elemento, devendo ser o primeiro *script* do código HTML, sendo que no *client-side* o próprio desenvolvedor deve incluí-lo, pois a biblioteca poderia ser executada no navegador depois de funções serem adicionadas na lista de eventos e essas funções não poderiam mais ser recuperadas pela biblioteca, já no *server-side* a biblioteca se encarrega disso; e adiciona os eventos correspondentes aos eventos não acessíveis via teclado que o elemento possui.

Os eventos listados na tabela de correspondência acima não possuem interação com outros elementos para o evento ser executado, o que faz com esses eventos correspondentes necessitem de um tratamento simples ou mesmo não necessitem para chamar seus eventos inacessíveis, como os eventos *keypress*, *keydown* e *keyup* que só devem chamar os eventos *click* ou *dblclick*, *mousedown* e *mouseup*, respectivamente, quando o usuário pressionar o Enter, e os eventos *focus* e *blur* que chamam os eventos *mouseover* e *mouseout*, respectivamente, sem nenhum tratamento.

Já os eventos de *Drag-and-Drop* possuem interação entre elementos, por isso sua solução é mais complexa. A solução implementada na biblioteca foi baseada no artigo de

James Edwards e no artigo de Gez Lemon, onde os mesmos apresentam formas alternativas para prover o acesso via teclado para o *Drag-and-Drop*. A biblioteca utilizou um conjunto de ações correspondentes via teclado para chamar os eventos de *Drag-and-Drop*, a ação correspondente do usuário para selecionar e arrastar o elemento pelo mouse via teclado se torna a ação do usuário focar no elemento e pressionar a tecla de espaço, assim o elemento é selecionado e são disparados os eventos correspondentes, caso haja outro elemento que tenha sido selecionado anteriormente o mesmo é liberado e o seu evento correspondente de liberação é disparado, além disso os elementos com os eventos *drop*, *dragenter*, *dragleave* e *dragover*, passam a possuir acesso via teclado, caso já não possuíssem, e é colocada a informação nesses elementos de qual será a ação que será realizada ao soltar o elemento sobre eles. Para liberar o elemento selecionado a qualquer momento o usuário deve pressionar a tecla Escape em qualquer parte da página, assim o seu evento correspondente de liberação é disparado e os elementos que possuem os eventos *drop*, *dragenter*, *dragleave* ou *dragover*, perdem o acesso via teclado, caso não possuíssem anteriormente. A ação de arrastar um elemento sobre outro elemento ou retirar um elementos arrastado sobre outro elemento, seja disparado um evento, possui como ação correspondente o usuário possuindo um elemento selecionado e foque sobre esse outro elemento ou foque e depois saía desse outro elemento, respectivamente. A ação de soltar um elemento sobre outro elemento, possui como ação correspondente o usuário possuindo um elemento selecionado e usuário pressione Enter sobre o outro elemento, assim o elemento selecionado é liberado e seu evento de liberação disparado, o evento de soltar o elemento sobre outro elemento é disparado e os elementos que possuem os eventos *drop*, *dragenter*, *dragleave* ou *dragover*, perdem o acesso via teclado, caso não possuíssem anteriormente. O HTML5 nativamente utiliza o recurso da área de transferência para trocar informações entre os elementos, mas não foi possível utilizar esse recurso nativo, pois ao ser desenvolvido o Google Chrome apresentava problemas ao utilizar a propriedade do objeto do evento, ao invés de utilizar objetos de eventos nativos foram utilizado objetos para simular os eventos nativos, isso permitiu que os eventos correspondentes no teclado possam ser utilizados no objeto que simula o evento.

Para evitar que para um evento inacessível de um elemento que já possui um evento correspondente seja chamado mais de uma vez ou substitua uma solução implementada pelo desenvolvedor, a biblioteca apenas adiciona os eventos correspondentes quando o

desenvolvedor não utilizou o evento correspondente ou quando a biblioteca já não utilizou o evento corresponde para chamar o mesmo evento, a biblioteca utiliza a sua rastreabilidade de soluções para verificar essa última situação, isso evita a redundância de métodos chamados como também dá preferência para a solução adotada pelo desenvolvedor.

### **3.2.2 Fornecer descrições para os campos**

O e-MAG recomenda a associação entre o campo e o seu rótulo através do atributo `for`. Por isso a biblioteca se encarrega de tentar resolver esse tipo de problema também, pois os leitores de tela podem possuir problema de leitura do rótulo do campo quando não são associados através do atributo `for`, mas o campo está como conteúdo do rótulo (BRASIL, 2009a).

### **3.2.3 Fornecer instruções para a entrada de dados**

O HTML5 nativamente possui atributos para os elementos que podem ser utilizados para fornecer mais informações de entrada de dados para o usuário, porém navegadores como o Internet Explorer 8 não fornecem essas informações para o leitor de tela que por sua vez consequentemente não fornece ao usuário. Sendo que a falta dessas informações podem causar um número maior de erros e dificuldade na utilização da página (SANTOS, 2011), por isso a biblioteca utiliza os estados e propriedades do WAI-ARIA correspondentes, além de modificar o atributo `aria-label`, para fornecer mais informações sobre o preenchimento de formulários para o usuário. O desenvolvedor pode utilizar a biblioteca em conjunto com o CSS para fornecer essas informações, caso o agente de usuário do usuário não possua suporte aos estados e propriedades do WAI-ARIA.

### **3.2.4 Web semântica e WAI-ARIA**

Os leitores de tela utilizam as informações semânticas e os atributos do ARIA para disponibilizar mais informações para as tecnologias assistivas. Porém modificar a *tag* dos

elementos pode causar diferenças visuais na página, por isso a biblioteca disponibiliza a possibilidade de adicionar ou alterar os atributos dos elementos, mas definir automaticamente os atributos e os valores dos atributos é um problema de conteúdo, por isso o desenvolvedor deve informar quais serão os seletores CSS que serão modificados e quais serão essas modificações, sendo próxima a solução utilizada por Eric Eggert.

### **3.2.5 Fornecer âncoras para ir direto a um bloco de conteúdo**

Essa solução corresponde a recomendação 1.5 do e-MAG 3.1 e a recomendação 2.4.1 do WCAG 2.0. Utilizar âncoras para ir diretamente para um bloco de conteúdo pode diminuir o tempo de navegação do usuário com deficiência visual, pois essa navegação geralmente é linear e esses links providos no início da página oferecem um acesso mais rápido para as partes de interesses do usuário (TAKAGI et al, 2007).

Definir quais os blocos de conteúdo que devem possuir um link de acesso direto para o conteúdo, é um problema de conteúdo, por isso é utilizado seletores CSS definidos pelo desenvolvedor para localizar e prover links de acesso direto para os elementos. O e-MAG 3.1 recomenda que esses links sejam utilizados em conjunto com teclas de atalho, baseado nisso a biblioteca também permite que seja associado aos links para ir direto a determinado conteúdo teclas de atalho, isso permite que em qualquer lugar da página o usuário ir direto para um bloco de conteúdo desde que conheça e utilize as teclas de atalho.

### **3.2.6 Associar células de dados às células de cabeçalho**

Essa solução corresponde a recomendação 3.10 do e-MAG 3.1 e recomendação 1.3.1 do WCAG 2.0. Ela ajuda a aumentar a acessibilidade quando a tabela é utilizada para apresentar dados tabulares, porém utilizar isso em tabelas utilizadas para layout pode piorar a acessibilidade da página, por isso a biblioteca realiza esse tratamento apenas em tabelas que estão sintaticamente e semanticamente correta, e com cabeçalho e corpo da tabela definido.

### 3.2.7 Exibir a lista de atalhos da página

Quando um elemento do código HTML possui alguma tecla de atalho definida, os navegadores como o Mozilla Firefox, Google Chrome, Safari e Internet Explorer não possuem um recurso nativo para informar ao usuário da existência dessas teclas de atalho mesmo dando suporte a esse recurso, já as versões do Opera anteriores a 15 ao pressionar as teclas ALT + Escape, exibe a lista de teclas de atalhos da página para o usuário ou então exibe uma mensagem informando que a página não possui teclas de atalhos.

A biblioteca procura proporcionar uma funcionalidade semelhante ao do Opera em todos os navegadores, mas com algumas diferenças tanto para garantir uma abrangência maior como para garantir que os objetivos da biblioteca sejam cumpridos.

Para corrigir esse problema foi percebido que para abrir algum dialogo como é feito no Opera seria necessário utilizar *scripts*, por isso a solução encontrada foi criar uma lista de atalhos que fica disponível na página, isso garante para o usuário que não será necessário ter algum conhecimento específico sobre o navegador, como é o caso do Opera, ou sobre a biblioteca para saber quais são os atalhos da página. A localização dessa lista pode ser definida pelo desenvolvedor ou no final da página que é o local padrão utilizado pela biblioteca, esse local foi escolhido para que o usuário que navega diversas vezes na mesma página não tenha que passar sempre pela lista de atalhos, apesar de ser possível ir para a lista diretamente através dos âncoras para ir direto a um bloco de conteúdo mencionadas anteriormente. Como cada navegador pode utilizar um conjunto de teclas de atalho distintos para gerar uma ação a biblioteca procura informar esse conjunto de teclas de atalho para o usuário, isso garante que o usuário não precise ter conhecimento prévio sobre como utilizar as teclas de atalho do navegador.

Outra diferença em relação ao Opera, é que o Opera utiliza o atributo de título do elemento para descrever o atalho, a biblioteca por outro lado utiliza os atributos ou o próprio conteúdo do elemento em último caso, para descrever o atalho. E no intuito de melhorar o recurso do Opera, caso o elemento não possua um atributo de título o mesmo será definido pela biblioteca, o que faz com que o usuário que utilize o recurso nativo do Opera possa se beneficiar do uso da biblioteca.

### **3.2.8 Prover uma forma de navegação para a descrição longa de um elemento não-textual**

Apesar de existir na especificação do elemento IMG o atributo de descrição longa da imagem (W3C, 2014b), foi percebido que os navegadores Google Chrome, Safari, Internet Explorer, Lynx e WebVox não possuem uma forma de navegar para a descrição longa das imagens, entretanto o Mozilla Firefox e o Opera oferecem no menu quando o elemento está focado a possibilidade de navegar para a descrição longa da imagem.

A biblioteca utilizou links para prover acesso às páginas da descrição longa utilizando o texto alternativo da imagem acompanhado do texto para definir descrições longas de imagens.

### **3.2.9 Promover a navegação entre os cabeçalhos da página**

O JAWS e o *Home Page Reader* (HPR) possibilitam a mudança na forma de navegação do usuário disponibilizando uma funcionalidade que os navegadores não disponibilizam nativamente, a possibilidade de navegação entre os cabeçalhos da página. Quando a página utiliza os cabeçalhos da página para separar as seções do página, além de melhorar a semântica da mesma, possibilita que o usuário possa navegar de forma mais eficiente, fazendo com que chegue mais rápido em uma seção da página, para os usuários com deficiência visual que utilizam leitores de tela, a forma de navegação é linear o que faz com que chegar em um determinado conteúdo da página possa ser um processo demorado (TAKAGI et al., 2007).

A biblioteca procura baseado no JAWS e no HPR, disponibilizar alguma forma de navegação entre os cabeçalhos da página. Tendo como exigência para promover essa forma de navegação a sintaxe correta dos cabeçalhos (h1-h6), pois não estando correta os elementos de cabeçalho podem estar sendo utilizados incorretamente para layout. A navegação entre os cabeçalhos da página utiliza um sumário da página, assim como é feito em um livro, onde esta disponível para o usuário uma lista de links que apontam para cada cabeçalho da página. Assim como a lista de teclas de atalhos esse sumário a localização desse sumário seria definida pelo desenvolvedor ou no final da página, para garantir que caso o usuário não deseje utilizar essa forma de navegação o mesmo não seja prejudicado por isso, também sendo

possível utilizar um link para apontar para o sumário. Apesar dessa ideia ter sido a escolhida para promover a navegação entre os cabeçalhos ela apresenta uma desvantagem se comparada com as demais, nela é necessário percorrer a ordem dos cabeçalhos.

## **4 HATEMILE FOR BROWSER**

O HaTeMiLe for Browser é nome utilizado para as extensões desenvolvidas para os navegadores Mozilla Firefox, Google Chrome e Opera, para que os navegadores utilizem o HaTeMiLe nas páginas visitadas pelo usuário, mesmo que os desenvolvedores da página não estejam utilizando a biblioteca. Apesar da diferença entre os navegadores para se desenvolver as extensões se decidiu utilizar o mesmo nome para que o usuário possa encontrar com facilidade a extensão para o seu navegador.

O HaTeMiLe for Browser apenas utiliza o HaTeMiLe para procurar resolver os problemas de acessibilidade na página, dessa forma os problemas não contemplados pelo HaTeMiLe não são resolvidos pelo HaTeMiLe for Browser.

A versão do HaTeMiLe for Browser para o Google Chrome está disponível em <<https://chrome.google.com/webstore/detail/hatemile-for-browser/cgmocdadonifbjlfaeedjdlhghkpfck>> e para o Opera está disponível em <<https://addons.opera.com/pt-br/extensions/details/hatemile-for-browser/>>, a versão do HaTeMiLe for Browser para o Mozilla Firefox ainda não foi aprovada.

### **4.1 Características de todas as extensões**

#### **4.1.1 Escolha das soluções executadas**

As extensões permitem que o usuário escolha quais serão as soluções implementadas pelo HaTeMiLe que serão executadas no navegador, com exceção da solução referente a Web semântica e ao WAI-ARIA que sempre são executadas pelos navegadores, por padrão as extensões executam todas as soluções do HaTeMiLe. As soluções são definidas através da tela de preferências nativa de cada navegador.



### **4.1.2 Visual da página**

As extensões permitem que o usuário escolha se haverá mudanças no visual de cada solução, serão ocultadas e disponíveis apenas para os leitores de tela, por padrão todas as mudanças no visual são ocultadas, a motivação para deixar todas as mudanças visuais ocultadas foi a possibilidade do usuário que utiliza o HaTeMiLe possa compartilhar o navegador com outra pessoa que não utilize um leitor de tela e possa se incomodar com as mudanças visuais e desabilitar a extensão.

### **4.1.3 Alterar textos**

As extensões permitem que o usuário possa definir os textos gerados pelas soluções da biblioteca, porém para que a biblioteca não dependa do usuário para informar os textos disponibilizados são definidos textos padrões que utilizam a internacionalização da extensão.

### **4.1.4 Internacionalização**

As extensões foram desenvolvidas de forma a permitir a internacionalização, tanto para os textos da tela de preferências da extensão como para os textos padrões gerados pelas soluções da biblioteca, sendo disponibilizadas em dois idiomas o inglês, idioma padrão das extensões, e o português do Brasil, foram utilizados recursos nativos dos navegadores para possuir a internacionalização, dessa forma o idioma utilizado pelo navegador também será utilizado pela extensão. As extensões também permitem que ao mudar o idioma do navegador mude também os textos padrões que a biblioteca utiliza, desde que o usuário não tenha modificado, tal comportamento foi uma sugestão da equipe de revisores das extensões do Mozilla.

### **4.1.5 Escopo das variáveis JavaScript**

As extensões foram desenvolvidas utilizando a versão HaTeMiLe em JavaScript que executa

no *client-side*, porém os navegadores Mozilla Firefox, Google Chrome e Opera permitem que sejam utilizados ambientes de execução diferentes para se executar o JavaScript (BAMBERG, 2014; GOOGLE, 2013), esses ambiente protegem as variáveis definidas nas extensões para que não estejam disponíveis para a página e vice-versa.

Ao serem desenvolvidas as extensões utilizavam o ambiente de execução da extensão, para evitar que as variáveis definidas pelas extensões pudessem substituir as variáveis definidas pelos desenvolvedores das páginas, porém em todos os navegadores foi verificado que não era possível verificar a existência de eventos JavaScript através dos atributos dos objetos no DOM ou na lista de eventos do objeto, dessa forma não era possível solucionar os problemas referentes a eventos que estão disponíveis apenas via mouse para torná-los disponíveis via teclado.

Para resolver esse problema e evitar conflitos com as variáveis da página, a biblioteca é executada no ambiente de execução da extensão, onde é realizado a configuração da extensão, são chamadas todas as soluções que serão executadas e são executadas as soluções que não são referentes aos eventos das páginas, já no ambiente de execução da página é executada apenas a solução referente aos eventos da página com o uso de *namespaces* para proteger as variáveis (MOZILLA, 2013a). Em versões anteriores das extensões todas as soluções eram executadas no ambiente de execução da página sem proteger as variáveis e métodos das extensões, porém por recomendação da equipe de revisores das extensões do Mozilla as variáveis e métodos inseridos pelas extensões passaram a ser protegidos utilizando o *namespace* e a maioria das soluções passaram a serem executadas no ambiente de execução da extensão.

Apesar de ser executado no *namespace* o jQuery sempre irá para o escopo global do ambiente de execução da página, por isso antes de utilizar o *script* do jQuery é verificado se a página já utiliza o jQuery incluindo o jQuery apenas se a página já não o utiliza, como algumas bibliotecas JavaScript utilizam a variável \$ (cifrão) ao incluir o jQuery na página é chamada a função que evita conflitos com outras bibliotecas mesmo que a variável \$ não esteja sendo utilizada (JQUERY, [201-]).

#### 4.1.6 Ícone

O navegador Opera exige que para que uma extensão seja submetida seja necessário que a mesma possua um ícone (BOVENS, [201-]), dessa forma foi criado um ícone baseado na própria documentação de boas práticas para criação de ícones (OPERA, [201-]), foi utilizado um plano de fundo laranja com a imagem de um cão guia encontrada na internet que está disponível sobre a licença *Creative Commons 0*, que permite que a imagem possa ser modificada e utilizada sem pedir autorização do autor. Apesar de apenas o Opera e o Chrome exigirem um ícone o mesmo também foi utilizado na extensão para o Mozilla Firefox.

Figura 2- Ícone do HaTeMiLe for Browser



#### 4.1.7 Conflito de identificadores

Para evitar conflito de identificadores na página caso a página também utilize o HaTeMiLe, foram modificados os prefixos dos identificadores para as extensões.

#### 4.1.8 jQuery

As extensões por padrão utilizam a versão 1.9.1 do jQuery, como também pode ser utilizado o jQuery da página independentemente da versão. Nas versões anteriores das extensões foram utilizadas a versão 1.6.4 com substituição da página caso o mesmo fosse uma versão anterior a 1.6.4, mas conforme alertado pela equipe de revisores das extensões do Mozilla, a estratégia adotada poderia causar falhas na página devido a utilização de *plug-ins* do jQuery e a versão utilizada apresentava uma falha de segurança que foi corrigida apenas nas versões posteriores

a 1.9 (JQUERY, 2013).

#### 4.1.9 *Frames*

Durante o desenvolvimento da extensão do Mozilla Firefox foi percebido que as páginas que utilizam o editor de texto rico TinyMCE, incluíam elementos e *scripts* dentro do conteúdo do TinyMCE, porém por se tratar de um editor de texto rico a inclusão de algum elemento fazia com que ao ser submetido, os elementos e *scripts* adicionados no conteúdo do TinyMCE fossem para a página de visualização desse conteúdo, o que pode causar problemas de usabilidade ou mesmo de acessibilidade na página, mesmo que os elementos adicionados fossem ocultados na extensão utilizado pelo usuário que submeteu o conteúdo, para os usuários que não utilizam a extensão iriam visualizar os elementos gerados pela extensão.

Foi verificado que isso acontecia, pois o TinyMCE utiliza o elemento IFRAME para gerar o editor de texto rico e o Mozilla Firefox considera o IFRAME do TinyMCE como uma nova página, dessa forma foi necessário forçar para que as extensões seja executada apenas no documento principal da página, isso significa que todas as páginas que forem executadas através de *frames* não terão seus problemas de acessibilidade resolvidos.

Os *frames* geralmente são utilizados pelas páginas para que sejam adicionadas propagandas ou alguma funcionalidade de uma rede social, apesar do usuário poder se beneficiar pela correção de alguns problemas de acessibilidade nos *frames*, para evitar causar problemas de usabilidade nas páginas mesmos em navegadores que não utilizam a extensão, as extensões não procuram corrigir os erros no TinyMCE ou outros editores de texto ricos.

Apesar dessa falha ter sido detectada apenas no Mozilla Firefox as extensões para o Google Chrome e Opera também procuram prevenir essa falha, desabilitando as soluções para os *frames*.

#### 4.1.10 **Alteração na biblioteca original**

Na documentação de desenvolvimento de extensões do navegador Mozilla Firefox é desaconselhado o uso da propriedade *innerHTML*, sendo que para ser utilizado em alguma

extensão deve ser utilizado algum método para não permitir que códigos HTML dinâmicos sejam inseridos (MOZILLA, 2015a), como em nenhuma solução implementada utiliza um método que utilize a propriedade *innerHTML* os métodos *getInnerHTML*, *setInnerHTML*, *getOuterHTML* e *cloneElement* foram removidos da classe *HTMLDOMElement* e de sua implementação nas extensões.

## 4.2 Extensão para o Mozilla Firefox

A extensão do Mozilla Firefox criada procura ser compatível com as versões 4.0 e superiores do navegador, dessa forma a extensão foi desenvolvida utilizando a tecnologia XUL que é implementada através de XML, JavaScript e CSS (MOZILLA, 2013b, 2014, 2015b).

Após desenvolvida a extensão para disponibilizá-la é necessário submeter a extensão para aprovação da equipe de revisores das extensões do Mozilla, quando a extensão é totalmente é aprovada na revisão completa a mesma é disponibilizada na página de extensões do Mozilla e é possível baixar a extensão através do próprio navegador, porém quando a revisão completa não é totalmente aprovada ou a extensão é aprovada na revisão parcial é dado um alerta ao usuário na página de extensões do Mozilla e não é possível baixar a extensão através do próprio navegador. Também é possível disponibilizar a extensão disponibilizando a extensão para download pelo próprio usuário, porém esse método não é confiável para o usuário e o mesmo é alertado sobre os riscos de instalar extensão (MOZILLA, 2015c). O HaTeMiLe for Browser foi submetida para aprovação da equipe de revisores do Mozilla, porém até o momento os mesmos não aprovaram a última versão extensão, as demais versões foram apagadas por conterem falhas graves.

## 5 HATEMILE FOR WP

O HaTeMiLe for WP é o nome do *plug-in* para o WordPress, um *Content Manager System* (CMS) utilizado para se criar sites e blogs, para que o WordPress utilize a biblioteca em suas páginas, o WordPress foi desenvolvido em PHP, assim a versão PHP da biblioteca foi utilizada.

Diferentemente do HaTeMiLe for Browser o *plug-in* é executado no *server-side* e é responsável apenas por executar todas as soluções da biblioteca, dessa forma a configuração dele deve ser realizada no arquivo XML de configuração do HaTeMiLe.

O HaTeMiLe for WP está disponível em <<https://wordpress.org/plugins/hatemile-for-wp/>>.

### 5.1 Extensão para o Google Chrome e a compatibilidade com o Opera

A extensão do Google Chrome diferente da extensão do Mozilla Firefox utiliza HTML, JavaScript e CSS, o JavaScript é executado através do ambiente *Content Scripts* que permite que o JavaScript possa manipular diretamente os elementos da página.

A partir da versão 15 o navegador Opera passou a utilizar a mesma arquitetura que as extensões do Google Chrome (DIXIT, [201-]; OPERA, 2015). Como não foram utilizados recursos específicos dos navegadores, foi possível utilizar a mesma extensão para ambos os navegadores.

As extensões para o Google Chrome necessitam que sejam especificadas no seu arquivo de configuração quais serão as permissões que a extensão poderá utilizar no navegador. Na extensão foi colocada apenas as permissões para que seja possível acessar e modificar todas as páginas Web e também a permissão necessária para ler e modificar as preferências do usuário no navegador.

Por utilizar HTML, CSS e JavaScript, para renderizar a página de preferências do navegador, foi utilizado o *framework front-end* Bootstrap em sua versão 3.0.0 para ter um visual melhor na página de configurações.

Assim como o Mozilla Firefox para disponibilizar uma extensão é necessário que a mesma seja disponibilizada nos seus respectivos sites de extensões e apesar de ser possível disponibilizar uma extensão no site de extensões do Opera e instalá-lo no Google Chrome e vice-versa, as extensões foram submetidas para o site de extensões de cada navegador. O Opera também possui sua equipe de revisores que aprovam se a extensão será disponibilizado no seu site de extensões, porém diferentemente das extensões do Mozilla a extensão é aprovada apenas totalmente. Já o Google Chrome disponibiliza a extensão em seu site de extensões em uma 1 hora sem realizar uma revisão.

## 6 TESTES REALIZADOS COM O HATEMILE

Foram realizados testes em cinco páginas diferentes, em diferentes navegadores e leitores de tela para mostrar os resultados conseguidos utilizando o HaTeMiLe nas páginas, isso porque cada navegador e leitor de tela pode possuir uma saída diferente para a mesma página (BRASIL, 2009b). A intenção desses testes é mostrar as diferenças entre as páginas que utilizam o HaTeMiLe e as páginas que não utilizam o HaTeMiLe.

No sistema operacional Windows foi utilizado o leitor de tela *NonVisual Desktop Access* (NVDA) 2015.3 nos navegadores Mozilla Firefox 40.0, Google Chrome 39.0.2171.71, Opera 31.0, Internet Explorer 9 e Safari 5.34.57.2, porém o NVDA não conseguiu ler as páginas abertas no Safari, esse leitor de tela possui a funcionalidade de exibir em texto tudo o que ele está falando, por isso os seus resultados para cada navegador e página estão no Apêndice A. No sistema operacional Linux foi utilizado o leitor de tela Orca 3.10.3 no navegador Mozilla Firefox, infelizmente não foi possível testar no Google Chrome e Opera, pois os mesmos não são reconhecidos pelo leitor de tela Orca (CHROMIUM, [20--]), porém diferentemente do NVDA a única forma do Orca salvar o que está sendo falado é através do *debug* que é verboso e não informa de forma simples o que o leitor de tela está falando, devido a isso foi gravado um vídeo ao invés do *debug* do Orca.

Nas cinco páginas, três correspondem a mesma página de teste onde são exploradas todas as funcionalidades do HaTeMiLe, onde uma delas não executa o HaTeMiLe, outra apenas executa o HaTeMiLe na versão JavaScript e outra executa o HaTeMiLe na versão JavaScript em conjunto com o CSS para forçar os navegadores a exibirem as informações geradas pelo HaTeMiLe, duas páginas são exclusivas para o teste de semântica onde uma delas não executa o HaTeMiLe e a outra que executa.

### 6.1 Disponibilizar todas as funções da página via teclado

Nos navegadores **Mozilla Firefox**, **Google Chrome** e **Opera** em nenhuma das páginas em que os elementos possuem eventos correspondentes via teclado o leitor de tela não informou esse recurso. Porém no navegador **Internet Explorer** é informado que os elementos com



eventos correspondentes via teclado é informado que se trata de um campo de entrada, o que pode confundir o usuário.

## 6.2 Fornecer descrições para os campos

Em todos os navegadores é informado corretamente a descrição para os campos em todas as páginas.

## 6.3 Fornecer instruções para a entrada de dados

Para os campos obrigatórios, nos navegadores **Mozilla Firefox utilizando o leitor de tela NVDA**, **Google Chrome** e **Opera** é informado que o campo é obrigatório em todas as páginas, sendo que a página que utiliza CSS torna a informação de campo obrigatório redundante. No navegador **Internet Explorer** na página que não utiliza o HaTeMiLe não foi informado que o campo é obrigatório, sendo informado apenas nas páginas que utilizam o HaTeMiLe, sendo que a página que utiliza o CSS não causou redundância, pois o conteúdo gerado pelo CSS foi ignorado. No navegador **Mozilla Firefox utilizando o leitor de tela Orca** apenas na página que utiliza o HaTeMiLe que utiliza o CSS foi informado que o campo é obrigatório.

Para os campos com intervalo de valores, nos navegadores **Mozilla Firefox utilizando o leitor de tela NVDA**, **Mozilla Firefox utilizando o leitor de tela Orca**, **Google Chrome** e **Opera** apenas a página que utiliza o HaTeMiLe em conjunto com CSS exibiu os valores mínimos e máximos dos campos, sendo que os navegadores também informaram que o campo se tratava de um botão de rotação em todas as páginas. No navegador **Internet Explorer** em todas as páginas não é informado os valores mínimos e máximos dos campos, pois o conteúdo gerado pelo CSS foi ignorado, sendo que apenas nas páginas que utilizam o HaTeMiLe é informado que o campo se tratava de um botão de rotação.

Para os campos com autocompletar, no navegador **Mozilla Firefox utilizando o leitor de tela NVDA**, em todas as páginas quando o campo possui a funcionalidade de autocompletar sendo redundante na página que utiliza o HaTeMiLe em conjunto com o CSS, porém apenas a

página que utiliza o HaTeMiLe em conjunto com o CSS informa quando o campo não possui a funcionalidade de autocompletar. Para os navegadores **Mozilla Firefox utilizando o leitor de tela Orca**, **Google Chrome** e **Opera** informa que o campo possui a funcionalidade de autocompletar apenas na página que utiliza o HaTeMiLe em conjunto com o CSS. Para o navegador **Internet Explorer** não há nenhuma diferença entre as páginas, pois o navegador ignora o conteúdo gerado pelo CSS.

#### **6.4 Web semântica e WAI-ARIA**

Nos navegadores **Mozilla Firefox utilizando o leitor de tela NVDA**, **Mozilla Firefox utilizando o leitor de tela Orca**, **Google Chrome** e **Opera**, não houve nenhuma diferença entre as páginas. No navegador **Internet Explorer** é informado que a área de navegação e a parte principal da página na página que utiliza o HaTeMiLe.

#### **6.5 Fornecer âncoras para ir direto a um bloco de conteúdo**

Em todos os navegadores são disponibilizados os link que saltam para os conteúdos da página nas páginas que utilizam o HaTeMiLe.

#### **6.6 Associar células de dados às células de cabeçalho**

Em todos os navegadores não houveram diferenças entre as páginas.

#### **6.7 Exibir a lista de atalhos da página**

Nos navegadores **Mozilla Firefox utilizando o leitor de tela NVDA**, **Google Chrome** e **Internet Explorer**, em todas as páginas é informado que os elementos possuem teclas de atalho, gerando redundância nas páginas que utilizam o HaTeMiLe. No navegador **Mozilla**

**Firefox utilizando o leitor de tela Orca**, não é informado que os elementos possuem teclas de atalho, apenas informando sobre os atalhos da página nas páginas que utilizam o HaTeMiLe. No navegador **Opera**, é informado que os elementos possuem teclas de atalho, gerando redundância na página que utiliza o HaTeMiLe, sendo que na página que utiliza o HaTeMiLe sem CSS a informação nativa sobre o atalhos desaparece.

#### **6.8 Prover uma forma de navegação para a descrição longa de um elemento não-textual**

No navegador **Mozilla Firefox utilizando o leitor de tela NVDA**, em todas as páginas informa que a imagem possui uma descrição longa, mas apenas disponibiliza a funcionalidade de selecionar de navegar para a descrição longa através do mouse, porém nas páginas que utilizam o HaTeMiLe é possível selecionar o link e navegar para página. No **Mozilla Firefox utilizando o leitor de tela Orca** não é informado que a imagem possui uma descrição longa, mas permite acessar a descrição longa através do mouse, porém nas páginas que utilizam o HaTeMiLe é possível selecionar o link para navegar para página. No navegador **Google Chrome** não é informado que a imagem possui uma descrição longa e não permite acessar a descrição nem pelo mouse, porém nas páginas que utilizam o HaTeMiLe é possível selecionar o link e navegar para página, sendo que na página que utiliza o CSS a informação do texto alternativo da imagem sumiu. No navegador **Opera e Internet Explorer**, não informa que a imagem possui uma descrição longa e não permite acessar a descrição nem pelo mouse, porém nas páginas que utilizam o HaTeMiLe é possível selecionar o link e navegar para página.

#### **6.9 Promover a navegação entre os cabeçalhos da página**

Nos navegadores **Mozilla Firefox utilizando o leitor de tela NVDA**, **Mozilla Firefox utilizando o leitor de tela Orca**, **Google Chrome**, **Opera** e **Internet Explorer**, são disponibilizados os links para os cabeçalhos da página nas páginas que utilizam o HaTeMiLe.

## 7 CONCLUSÃO

A acessibilidade na web possui muitas ferramentas capazes de auxiliar os desenvolvedores no desenvolvimento de páginas mais acessíveis, porém a maioria delas apenas presta informações para os desenvolvedores sobre o que deve ser feito para melhorar a acessibilidade das páginas desenvolvidas, o que nem sempre o desenvolvedor pode realizar devido ao uso de *frameworks*, nesse sentido o HaTeMiLe além de procurar ser uma biblioteca que auxilia ao desenvolvedor que respeita os padrões da Web a tornar a página mais acessível alterando o código HTML da página, também procura estimular o desenvolvimento de softwares que alterem o código HTML das páginas.

Ainda que o presente trabalho apresente um software capaz de melhorar a acessibilidade na Web foi percebido que o mesmo não consegue fazer-se cumprir a maioria das recomendações do WCAG 2.0 e do eMAG 3.1. Também foram percebidas as diferenças entre os leitores de tela e navegadores, com informações que são ocultadas para usuário nesse sentido é interessante também o desenvolvimento de um software capaz de exibir para o usuário todas as informações úteis para o mesmo.

## REFERÊNCIAS

- AKQUINET. **The state of web accessibility in the JavaEE world**. 2013. Disponível em: <http://blog.akquinet.de/2013/07/26/the-state-of-web-accessibility-in-the-javaee-world/>. Acessado em 01/01/2015.
- APACHE. **Apache License v2.0 and GPL Compatibility**. [2007?]. Disponível em: <http://www.apache.org/licenses/GPL-compatibility.html>. Acessado em 01/01/2015.
- \_\_\_\_\_. **ASF Legal Previously Asked Questions**. [2004?]. Disponível em: <http://apache.org/legal/resolved.html>. Acessado em: 07/02/2015.
- BAMBERG, Will. **Changes to unsafeWindow for the Add-on SDK. Mozilla Add-ons Blog**. 2014. Disponível em: <https://blog.mozilla.org/addons/2014/04/10/changes-to-unsafeWindow-for-the-add-on-sdk/#comments>. Acessado em 10/11/2015.
- BOVENS, Andreas. **Publishing guidelines**. Opera. [201-]. Disponível em: [https://dev.opera.com/extensions/tut\\_publishing\\_guidelines.html](https://dev.opera.com/extensions/tut_publishing_guidelines.html). Acessado em: 10/11/2015.
- BRASIL. **Modelo de acessibilidade de governo eletrônico (e-MAG)**. Brasília, DF, 2014.
- \_\_\_\_\_. **Leitores de Tela - Pontos de Fragilidade**. Brasília, DF, 2009a.
- \_\_\_\_\_. **Leitores de Tela - Descrição e Comparativo**. Brasília, DF, 2009b.
- CASS, Stephen. **Top 10 Programming Languages**. IEEE Spectrum. 2014. Disponível em: <http://spectrum.ieee.org/computing/software/top-10-programming-languages>. Acessado em 01/01/2015.
- CENTENO, Vicente Luque et al. Web accessibility evaluation tools: A survey and some improvements. **Electronic notes in theoretical computer science**, v. 157, n. 2, p. 87-100, 2006.
- CHALEGRE, Virgínia Carvalho. **Uma metodologia de testes de acessibilidade para usuários cegos em ambientes web**. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática da Universidade Federal de Pernambuco. Recife: Universidade Federal de Pernambuco, 2011.
- CHROMIUM. **Accessibility: Assistive technology support**. [20--]. Disponível em: <https://www.chromium.org/user-experience/assistive-technology-support>. Acessado

em: 14/11/2015.

COFFEESCRIPT. **CoffeeScript**. [2014?]. Disponível em: <<http://coffeescript.org/>>. Acessado em: 10/02/2015.

DIXIT, Shwetank. **The basics of making an extension**. [201-]. Opera. Disponível em: <[https://dev.opera.com/extensions/tut\\_basics.html](https://dev.opera.com/extensions/tut_basics.html)>. Acessado em: 10/11/2015.

FINLEY, Klint. **5 Ways to Tell Which Programming Languages are Most Popular**. ReadWrite. 2012. Disponível em: <<http://readwrite.com/2012/06/05/5-ways-to-tell-which-programming-lanugages-are-most-popular>>. Acessado em: 09/01/2015.

EDWARDS, James. **Accessible Drag and Drop with Multiple Items**. sitepoint. 2015. Disponível em: <<http://www.sitepoint.com/accessible-drag-drop/>>. Acessado em: 16/06/2015.

EGGERT, Eric. **accessifyhtml5.js**. 2014. Disponível em: <<https://github.com/yatil/accessifyhtml5.js>>. Acessado em: 28/11/2015.

GITHUB. **Features**. [20--]. Disponível em: <<https://github.com/features/>>. Acessado em 01/01/2015.

GOOGLE. **Content Scripts**. 2013. Disponível em: <[https://developer.chrome.com/extensions/content\\_scripts](https://developer.chrome.com/extensions/content_scripts)>. Acessado em 10/11/2015.

GUERCIO, Angela et al. Addressing challenges in web accessibility for the blind and visually impaired. **System and Technology Advancements in Distance Learning**, p. 249, 2012.

HOCK-CHUAN, Chua. **Object-Oriented Programming (OOP) in C++**. 2013. Disponível em: <[https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp3\\_OOP.html](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp3_OOP.html)>. Acesso em: 10/02/2015.

JQUERY. **jQuery.noConflict()**. [201-]. Disponível em: <<https://api.jquery.com/jquery.noconflict/>>. Acessado em: 10/11/2015.

\_\_\_\_\_. **jQuery Core 1.9 Upgrade Guide**. 2013. Disponível em: <<http://jquery.com/upgrade-guide/1.9/>>. Acessado em: 10/11/2015.

LEMON, Gez. **Accessible Drag and Drop Using WAI-ARIA**. Opera. 2009. Disponível em: <<https://dev.opera.com/articles/accessible-drag-and-drop/>>. Acessado em: 15/06/2015.

LOIACONO, Eleanor T.; DJAMASBI, Soussan; KIRYAZOV, Todor. Factors that affect visually impaired users' acceptance of audio and music websites. **International Journal**

**of Human-Computer Studies**, v. 71, n. 3, p. 321-334, 2013.

MOZILLA. **JavaScript Object Management**. 2013a. Disponível em:

<[https://developer.mozilla.org/en-US/Add-](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/JavaScript_Object_Management)

[ons/Overlay\\_Extensions/XUL\\_School/JavaScript\\_Object\\_Management](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/JavaScript_Object_Management)>. Acessado em: 10/11/2015.

\_\_\_\_\_. **Introduction**. 2013b. Disponível em: <[https://developer.mozilla.org/en-US/Add-](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/Introduction)

[ons/Overlay\\_Extensions/XUL\\_School/Introduction](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/Introduction)>. Acessado em: 10/11/2015.

\_\_\_\_\_. **XUL**. 2014. Disponível em: <[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL)

[US/docs/Mozilla/Tech/XUL](https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL)>. Acessado em: 10/11/2015.

\_\_\_\_\_. **Appendix E: DOM Building and Insertion (HTML & XUL)**. 2015a. Disponível em: <[https://developer.mozilla.org/en-US/Add-](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/DOM_Building_and_HTML_Insertion)

[ons/Overlay\\_Extensions/XUL\\_School/DOM\\_Building\\_and\\_HTML\\_Insertion](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions/XUL_School/DOM_Building_and_HTML_Insertion)>.

Acessado em: 10/11/2015.

\_\_\_\_\_. **Legacy extensions**. 2015b. Disponível em: <[https://developer.mozilla.org/en-](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions)

[US/Add-ons/Overlay\\_Extensions](https://developer.mozilla.org/en-US/Add-ons/Overlay_Extensions)>. Acessado em: 10/11/2015.

\_\_\_\_\_. **Review Policies**. 2015c. Disponível em: <[https://developer.mozilla.org/en-US/Add-](https://developer.mozilla.org/en-US/Add-ons/AMO/Policy/Reviews)

[ons/AMO/Policy/Reviews](https://developer.mozilla.org/en-US/Add-ons/AMO/Policy/Reviews)>. Acessado em: 10/11/2015.

OPERA. **Creating effective icons**. [201-]. Disponível em:

<[https://dev.opera.com/extensions/tut\\_icons.html](https://dev.opera.com/extensions/tut_icons.html)>. Acessado em: 10/11/2015.

\_\_\_\_\_. **Welcome!**. 2015. Disponível em: <<https://dev.opera.com/extensions/>>. Acessado em: 10/11/2015.

QUEIROZ, Marco Antonio de. **A Importância dos Padrões Web para a Acessibilidade de Sites**. 2009. Disponível em: <[http://www.acessibilidadelegal.com/23-padroes-](http://www.acessibilidadelegal.com/23-padroes-web.php)

[web.php](http://www.acessibilidadelegal.com/23-padroes-web.php)>. Acessado em: 23/12/2015.

ROCHA, Janicy Aparecida Pereira; DUARTE, Adriana Bogliolo Sirihal. **(In)Acessibilidade na web para pessoas com deficiência visual**: Um estudo de usuários à luz da cognição situada. In: XIV Encontro Nacional de Pesquisa em Ciência da Informação. 2013a.

\_\_\_\_\_. **Diretrizes de acessibilidade web**: um estudo comparativo entre as WCAG 2.0 e o e-MAG 3.0. In: Inc. Soc., Brasília, DF, v. 5 n. 2, p.73-86, jan./jun. 2012.

\_\_\_\_\_. O comportamento de usuários cegos durante o acesso mediado por leitores de tela: Um estudo sob o enfoque da cognição situada. **Perspectivas em Gestão &**

**Conhecimento**, v. 3, n. 3, p. 173-196, 2013b.

ROTARD, Martin; TARAS, Christiane; ERTL, Thomas. **Tactile web browsing for blind people**. 2007.

SANTOS, Danielle de Souza; PORTO, Marcelo Franco. **A acessibilidade da web frente às necessidades dos usuários que não podem visualizar o seu conteúdo**. 2011.

TAKAGI, Hironobu et al. Analysis of navigability of Web applications for improving blind usability. **ACM Transactions on Computer-Human Interaction (TOCHI)**, v. 14, n. 3, p. 13, 2007.

W3C. **Contacting Organizations about Inaccessible Websites**. 2010. Disponível em:

<<http://www.w3.org/WAI/users/inaccessible>>. Acessado em: 08/02/2015.

\_\_\_\_\_. **Web Content Accessibility Guidelines (WCAG) Overview**. 2012. Disponível em:

<<http://www.w3.org/WAI/intro/wcag.php>>. Acesso em: 10/02/2015.

\_\_\_\_\_. **Scripting – HTML5**. 2014a. Disponível em: <<http://www.w3.org/TR/2014/REC-html5-20141028/scripting-1.html>>. Acesso em: 10/02/2015.

\_\_\_\_\_. **HTML5 Image Description Extension (longdesc)**. 2014b. Disponível em:

<<http://www.w3.org/TR/2014/PR-html-longdesc-20141204/>>. Acesso em: 10/02/2015.

W3TECHS. **Usage of server-side programming languages for websites**. 2014. Disponível em: <[http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)>. Acessado em 01/01/2015.



## APÊNDICE A – SAÍDA EM TEXTO DO NVDA

Para mostrar a eficácia do HaTeMiLe foram desenvolvidas cinco páginas, três que correspondem a mesma página de teste onde são exploradas todas as funcionalidades do HaTeMiLe, sendo que uma página não executa o HaTeMiLe, outra página apenas executa o HaTeMiLe na versão JavaScript e outra página executa o HaTeMiLe na versão JavaScript em conjunto com o CSS para forçar os navegadores a exibirem as informações geradas pelo HaTeMiLe, as outras duas páginas são exclusivas para o teste de semântica onde uma delas não executa o HaTeMiLe e a outra que executa. A seguir as saídas em texto do leitor de tela NVDA das páginas para os navegadores Mozilla Firefox, Google Chrome, Opera e Internet Explorer.

### Página que testa todas as funcionalidades do HaTeMiLe, que não utiliza o HaTeMiLe

Código-fonte da página que testa todas as funcionalidades do HaTeMiLe, porém não utiliza o HaTeMiLe:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HaTeMiLe Tests</title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h1>HaTeMiLe Tests</h1>
    <!-- Events -->
    <div>
      <h2>Test Events</h2>
      <a href="#" onclick="alert('Alert A')">Alert</a>
      <button onclick="alert('Alert Button')">Alert</button>
      <input type="button" onclick="alert('Alert Input')
value="Alert" />
      <span onclick="alert('Alert Span')" style="background:
red;">Alert</span>
      <i onclick="alert('Alert I')">Alert</i>
      <div style="height: 300px; width: 300px; border: 1px solid black"
onclick="alert('Alert Div')">
        Alert
```

```

        </div>
        <span onclick="alert('Alert span')"
onkeypress="console.log('Console SPAN')" style="background:
blueviolet;">Console</span>
        <hr />
        <a href="#" onmouseover="console.log('Over A')"
onmouseout="console.log('Out A')">Console</a>
        <button onmouseover="console.log('Over Button')"
onmouseout="console.log('Out Button')">Console</button>
        <input type="button" onmouseover="console.log('Over Input')"
value="Console" onmouseout="console.log('Out Input')" />
        <span onmouseover="console.log('Over Span')"
style="background: red;" onmouseout="console.log('Out
Span')">Console</span>
        <i onmouseover="console.log('Over I')"
onmouseout="console.log('Out I')">Console</i>
        <div style="height: 300px; width: 300px; border: 1px solid black"
ondrop="event.preventDefault();event.target.appendChild(document.getE
lementById(event.dataTransfer.getData('text')));"
ondragover="event.preventDefault();">
            Console
        </div>
        <span id="draggable-item"
ondragstart="event.dataTransfer.setData('text', event.target.id);"
draggable="true">Drag-and-Drop</span>
    </div>
    <!-- Forms -->
    <form autocomplete="off" id="form1">
        <h2>Test Forms</h2>
        <label for="field1">Field1</label>
        <input type="text" value="" required="required" id="field1"
autocomplete="on" />
        <label>
            Field2
        <div>
            <input type="text" value="" required="required"
autocomplete="off" />
        </div>
        </label>
        <label for="field3">Field3</label>
        <textarea required="required" id="field3"
autocomplete></textarea>
        <label>
            Field4
            <textarea required="required"
autocomplete="none"></textarea>

```

```

</label>
<label for="field5">Field5</label>
<select required="required" id="field5">
  <option value="">0</option>
  <option value="1">1</option>
  <option value="2">2</option>
</select>
<label>
  Field6
  <select required="required">
    <option value="">0</option>
    <option value="1">1</option>
    <option value="2">2</option>
  </select>
</label>
<label for="field7">Field7</label>
<input type="number" min="0" value="0" max="10"
id="field7" />
<input type="submit" value="Submit" />
</form>
<input type="text" value="" required="" form="form1" />
<!-- Images -->
<div>
  <h2>Test Images</h2>
  
  
</div>
<!-- Shortcuts -->
<form action="http://www.webplatform.org/">
  <h2>Test Shortcuts</h2>
  <a href="http://www.w3.org/html/" title="Go to HTML5"
accesskey="q">HTML5</a><br />
  <a href="https://www.wikimedia.org/" accesskey="w">Go to
Wikimidia</a><br />
  <label id="label1">Field1</label>
  <input type="text" value="" aria-labelledby="label1"
accesskey="e" /><br />
  <input type="text" value="" aria-label="Field 2" accesskey="r"
/><br />
  <input type="image" src="octobiwan.jpg" alt="Octobiwan"
accesskey="t" /><br />
  <input type="reset" value="Reset button" accesskey="y"
/><br />
  <input type="button" value="Show shortcuts" accesskey="u"

```

```

onclick="alert('Only in client-side version.');" /><br />
    <input type="submit" value="Subit Button" accesskey="i"
/><br />
    </form>
    <!-- Selectors -->

    <!-- Tables -->
    <div>
        <h2>Test Tables</h2>
        <table>
            <thead>
                <tr>
                    <th rowspan="3">1</th>
                    <th rowspan="2">2</th>
                    <th>3</th>
                    <td>4</td>
                </tr>
                <tr>
                    <th colspan="2">1</th>
                </tr>
                <tr>
                    <th>1</th>
                    <th>2</th>
                    <td>3</td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr>
                    <td rowspan="2"></td>
                    <td></td>
                    <th></th>
                    <td></td>
                </tr>
                <tr>
                    <td colspan="2"></td>
                    <td></td>
                </tr>
                <tr>
                    <td></td>
                    <td></td>
                </tr>
            </tbody>
        </table>
    </div>

```

```

        <td></td>
      </tr>
    </tbody>
  </table>
  <table>
    <tr>
      <th>1</th>
      <td>2</td>
      <th colspan="2">3</th>
    </tr>
    <tr>
      <td>1</td>
      <td colspan="2">2</td>
      <td>3</td>
    </tr>
    <tr>
      <td>1</td>
      <th>2</th>
      <td>3</td>
      <td>4</td>
    </tr>
  </table>
</div>
</body>
</html>

```

Saída em texto para o navegador Mozilla Firefox:

```

HaTeMiLe Tests - Mozilla Firefox
HaTeMiLe Tests
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert

botão
Alert

botão
Alert

clícável

```

Alert

clicável

Alert

clicável

Alert

clicável

Console

separador

linque

Console

botão

Console

botão

Console

Console Console

Console

Drag-and-Drop

cabeçalho nível 2

Test Forms

Field1

edição entrada inválida exigido

Field2

edição entrada inválida exigido

Field3

edição entrada inválida exigido multilinha

fora de edição

Field4

edição entrada inválida exigido multilinha

fora de edição

Field5

caixa de combinação recolhido entrada inválida exigido

0

Field6

caixa de combinação recolhido entrada inválida exigido

0

Field7

botão de rotação  
edição  
0

botão  
Submit  
edição entrada inválida exigido  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Accessibility  
gráfico Tem descrição longa  
HTML5  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+Shift+q  
HTML5

linque Alt+Shift+w  
Go to Wikimidia

Field1

edição possui autocompletar Alt+Shift+e

edição possui autocompletar Alt+Shift+r  
botão Alt+Shift+t  
Octobiwan

botão Alt+Shift+y  
Reset button

botão Alt+Shift+u  
Show shortcuts

botão Alt+Shift+i

Subit Button

cabeçalho nível 2

Test Tables

tabela com 7 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

coluna 4

4

2 linha 2 coluna 3

1

linha 3 coluna 2

1

1 coluna 3

2

1 coluna 4

3

linha 4 1 coluna 1

1 coluna 2

1 coluna 3

1 coluna 4

linha 5 1 coluna 1

1 coluna 2

1 coluna 3

1 coluna 4

linha 6 1 coluna 2

1 coluna 4

linha 7 1 coluna 1

1 coluna 2

1 coluna 3



fora de tabela  
tabela com 3 linhas e 4 colunas  
linha 1 coluna 1  
1  
coluna 2  
2  
coluna 3  
3  
linha 2 coluna 1  
1  
coluna 2  
2  
3 coluna 4  
3  
linha 3 coluna 1  
1  
coluna 2  
2  
3 coluna 3  
3  
3 coluna 4  
4

Saída em texto para o navegador Google Chrome:

HaTeMiLe Tests documento  
desconhecido  
HaTeMiLe Tests  
cabeçalho nível 1  
HaTeMiLe Tests  
cabeçalho nível 2  
Test Events  
linque  
Alert  
botão  
Alert  
botão  
Alert  
AlertAlert  
Alert  
Console  
linque  
Console  
botão

Console  
botão  
Console  
ConsoleConsole  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1  
edição entrada inválida exigido

Field2  
edição entrada inválida exigido

Field3  
edição entrada inválida exigido multilinha  
fora de edição

Field4  
edição entrada inválida exigido multilinha  
fora de edição

Field5  
caixa de combinação recolhido entrada inválida exigido  
0

Field6  
caixa de combinação recolhido entrada inválida exigido  
0

Field7  
botão de rotação editável  
0  
botão de rotação editável  
botão de rotação editável  
botão editável

botão editável

botão  
Submit  
edição entrada inválida exigido  
cabeçalho nível 2  
Test Images  
gráfico  
Accessibility  
gráfico  
HTML5  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+q

## HTML5

linque Alt+w  
Go to Wikimidia

Field1  
edição Alt+e

edição Alt+r  
botão Alt+t  
Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 4 colunas  
linha 1 1  
1  
1 coluna 1  
1  
2  
2  
1 coluna 2  
2  
3  
1  
2 coluna 3  
3  
1 coluna 4  
4

linha 2 3

1

2 coluna 3

1

linha 3 2

2

1 coluna 2

1

3

1

2 coluna 3

2

1 coluna 4

3

linha 4 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1

1 coluna 4

3

1

2 coluna 3

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1

1 coluna 4

3

1

2 coluna 3

1

1 coluna 1

2

1 coluna 2

1 coluna 4

linha 7 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

fora de tabela

1

2

3

1

2

3

1

2

3

4

Saída em texto para o navegador Opera:

HaTeMiLe Tests documento

cabeçalho nível 1

HaTeMiLe Tests

HaTeMiLe Tests

cabeçalho nível 1

HaTeMiLe Tests

cabeçalho nível 2

Test Events

linque

Alert

botão

Alert

botão

Alert

AlertAlert

Alert

Console  
linque  
Console  
botão  
Console  
botão  
Console  
ConsoleConsole  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1  
edição entrada inválida exigido  
  
Field2  
edição entrada inválida exigido  
Field3  
edição entrada inválida exigido multilinha  
fora de edição  
Field4  
edição entrada inválida exigido multilinha  
fora de edição  
Field5  
caixa de combinação recolhido entrada inválida exigido  
0  
Field6  
caixa de combinação recolhido entrada inválida exigido  
0  
Field7  
botão de rotação editável  
0  
botão de rotação editável  
botão de rotação editável  
botão editável  
  
botão editável  
  
botão  
Submit  
edição entrada inválida exigido  
cabeçalho nível 2  
Test Images  
gráfico  
Acessibility  
gráfico

HTML5  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+q  
HTML5

linque Alt+w  
Go to Wikimidia

Field1  
edição Alt+e

edição Alt+r  
botão Alt+t  
Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 4 colunas  
linha 1 1  
1  
1 coluna 1  
1  
2  
2  
1 coluna 2  
2  
3  
1

2 coluna 3

3

1 coluna 4

4

linha 2 3

1

2 coluna 3

1

linha 3 2

2

1 coluna 2

1

3

1

2 coluna 3

2

1 coluna 4

3

linha 4 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 5 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4



```

linha 6 2
2
1 coluna 2

1 coluna 4

linha 7 1
1
1 coluna 1

2
2
1 coluna 2

3
1
2 coluna 3

fora de tabela
1
2
3
1
2
3
1
2
3
4

```

Saída em texto para o navegador Internet Explorer:

```

HaTeMiLe Tests - Internet Explorer
HaTeMiLe Tests documento
cabeçalho nível 1
HaTeMiLe Tests
HaTeMiLe Tests
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert
botão

```

Alert  
botão  
Alert  
clicável  
Alert  
clicável  
Alert  
clicável  
Alert  
clicável  
Console  
separador  
linque  
Console  
botão  
Console  
botão  
Console  
Console  
Console  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1  
edição  
  
Field2  
edição  
Field3  
edição multilinha  
fora de edição  
Field4  
edição multilinha  
fora de edição  
Field5  
caixa de combinação recolhido  
0  
Field6  
caixa de combinação recolhido  
0  
Field7  
edição  
0  
botão  
Submit

edição  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Acessibility  
gráfico Tem descrição longa  
HTML5  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+q  
HTML5

linque Alt+w  
Go to Wikimidia

Field1  
edição Alt+e

edição Alt+r  
botão Alt+t  
Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 3 colunas  
linha 1 coluna 1  
1  
coluna 2  
2

coluna 3

3

coluna 4

4

1 linha 2 3 coluna 3

1

1 linha 3 2 coluna 2

1

3 coluna 3

2

coluna 4

3

linha 4 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 5 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 6 2 coluna 2

coluna 4

linha 7 1 coluna 1

2 coluna 2

3 coluna 3

fora de tabela

tabela com 3 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

```

linha 2 1 coluna 1
1
coluna 2
2
3 coluna 4
3
linha 3 1 coluna 1
1
coluna 2
2
3 coluna 3
3
3 coluna 4
4

```

### **Página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe**

Código-fonte da página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe:

```

<!DOCTYPE html>
<html>
  <head>
    <title>HaTeMiLe Tests</title>
    <meta charset="UTF-8" />
  </head>
  <body>
    <h1>HaTeMiLe Tests</h1>
    <!-- Events -->
    <div>
      <h2>Test Events</h2>
      <a href="#" onclick="alert('Alert A')">Alert</a>
      <button onclick="alert('Alert Button')">Alert</button>
      <input type="button" onclick="alert('Alert Input')
value="Alert" />
      <span onclick="alert('Alert Span')" style="background:
red;">Alert</span>
      <i onclick="alert('Alert I')">Alert</i>
      <div style="height: 300px; width: 300px; border: 1px solid black"
onclick="alert('Alert Div')">
        Alert
      </div>
      <span onclick="alert('Alert span')
onkeypress="console.log('Console SPAN')" style="background:

```

```

blueviolet;">Console</span>
    <hr />
    <a href="#" onmouseover="console.log('Over A')"
onmouseout="console.log('Out A')">Console</a>
    <button onmouseover="console.log('Over Button')"
onmouseout="console.log('Out Button')">Console</button>
    <input type="button" onmouseover="console.log('Over Input')"
value="Console" onmouseout="console.log('Out Input')"/>
    <span onmouseover="console.log('Over Span')"
style="background: red;" onmouseout="console.log('Out
Span')">Console</span>
    <i onmouseover="console.log('Over I')"
onmouseout="console.log('Out I')">Console</i>
    <div style="height: 300px; width: 300px; border: 1px solid black"
ondrop="event.preventDefault();event.target.appendChild(document.getE
lementById(event.dataTransfer.getData('text')));"
ondragover="event.preventDefault();">
        Console
    </div>
    <span id="draggable-item"
ondragstart="event.dataTransfer.setData('text', event.target.id);"
draggable="true">Drag-and-Drop</span>
</div>
<!-- Forms -->
<form autocomplete="off" id="form1">
    <h2>Test Forms</h2>
    <label for="field1">Field1</label>
    <input type="text" value="" required="required" id="field1"
autocomplete="on"/>
    <label>
        Field2
    </div>
        <input type="text" value="" required="required"
autocomplete="off"/>
    </div>
</label>
    <label for="field3">Field3</label>
    <textarea required="required" id="field3"
autocomplete></textarea>
    <label>
        Field4
        <textarea required="required"
autocomplete="none"></textarea>
    </label>
    <label for="field5">Field5</label>
    <select required="required" id="field5">

```

```

        <option value="">0</option>
        <option value="1">1</option>
        <option value="2">2</option>
    </select>
    <label>
        Field6
        <select required="required">
            <option value="">0</option>
            <option value="1">1</option>
            <option value="2">2</option>
        </select>
    </label>
    <label for="field7">Field7</label>
    <input type="number" min="0" value="0" max="10"
id="field7" />
    <input type="submit" value="Submit" />
</form>
<input type="text" value="" required="" form="form1" />
<!-- Images -->
<div>
    <h2>Test Images</h2>
    
    
</div>
<!-- Shortcuts -->
<form action="http://www.webplatform.org/">
    <h2>Test Shortcuts</h2>
    <a href="http://www.w3.org/html/" title="Go to HTML5"
accesskey="q">HTML5</a><br />
    <a href="https://www.wikimedia.org/" accesskey="w">Go to
Wikimidia</a><br />
    <label id="label1">Field1</label>
    <input type="text" value="" aria-labelledby="label1"
accesskey="e" /><br />
    <input type="text" value="" aria-label="Field 2" accesskey="r"
/><br />
    <input type="image"
src="https://octodex.github.com/images/octobiwan.jpg" alt="Octobiwan"
accesskey="t" /><br />

```

```

        <input type="reset" value="Reset button" accesskey="y"
/><br />
        <input type="button" value="Show shortcuts" accesskey="u"
onclick="alert('Only in client-side version.');" /><br />
        <input type="submit" value="Subit Button" accesskey="i"
/><br />
    </form>
    <!-- Selectors -->

    <!-- Tables -->
    <div>
        <h2>Test Tables</h2>
        <table>
            <thead>
                <tr>
                    <th rowspan="3">1</th>
                    <th rowspan="2">2</th>
                    <th>3</th>
                    <td>4</td>
                </tr>
                <tr>
                    <th colspan="2">1</th>
                </tr>
                <tr>
                    <th>1</th>
                    <th>2</th>
                    <td>3</td>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td></td>
                    <td></td>
                    <td></td>
                    <td></td>
                </tr>
                <tr>
                    <td rowspan="2"></td>
                    <td></td>
                    <th></th>
                    <td></td>
                </tr>
                <tr>
                    <td colspan="2"></td>
                    <td></td>
                </tr>
            </tbody>
        </table>
    </div>

```



```

        <tr>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </tbody>
</table>
<table>
    <tr>
        <th>1</th>
        <td>2</td>
        <th colspan="2">3</th>
    </tr>
    <tr>
        <td>1</td>
        <td colspan="2">2</td>
        <td>3</td>
    </tr>
    <tr>
        <td>1</td>
        <th>2</th>
        <td>3</td>
        <td>4</td>
    </tr>
</table>
</div>
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/common.js"></script>
    <script type="text/javascript" src="js/config.js"></script>
    <script type="text/javascript"
src="js/eventlistener.js"></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleEventImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleFormImplementation.js"></sc
ript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleImageImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleNavigationImplementation.js"
></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleSelectorImplementation.js"><
/script>

```

```

        <script type="text/javascript"
src="js/hatemile/implementation/AccessibleTableImplementation.js"></sc
ript>
        <script type="text/javascript"
src="js/hatemile/util/CommonFunctions.js"></script>
        <script type="text/javascript"
src="js/hatemile/util/Configure.js"></script>
        <script type="text/javascript"
src="js/hatemile/util/SelectorChange.js"></script>
        <script type="text/javascript"
src="js/hatemile/util/Skipper.js"></script>
        <script type="text/javascript"
src="js/hatemile/util/jqueryandvanilla/JQueryHTMLDOMParser.js"></script
>
        <script type="text/javascript"
src="js/hatemile/util/jqueryandvanilla/VanillaHTMLDOMElement.js"></scri
pt>
        <script type="text/javascript">
            var configuration = new hatemile.util.Configure(config);
            var parser = new
hatemile.util.jqueryandvanilla.JQueryHTMLDOMParser(document);

            var accessibleEvent = new
hatemile.implementation.AccessibleEventImplementation(parser,
configuration);

            accessibleEvent.fixDragsandDrops();
            accessibleEvent.fixHovers();
            accessibleEvent.fixActives();

            var accessibleForm = new
hatemile.implementation.AccessibleFormImplementation(parser,
configuration);

            accessibleForm.fixRequiredFields();
            accessibleForm.fixRangeFields();
            accessibleForm.fixAutoCompleteFields();
            accessibleForm.fixLabels();

            var accessibleImage = new
hatemile.implementation.AccessibleImageImplementation(parser,
configuration);

            accessibleImage.fixLongDescriptions();

            var accessibleNavigation = new
hatemile.implementation.AccessibleNavigationImplementation(parser,
configuration, navigator.userAgent);
            accessibleNavigation.fixHeadings();

```

```

        accessibleNavigation.fixShortcuts();
        accessibleNavigation.fixSkippers();

        var accessibleSelectors = new
hatemile.implementation.AccessibleSelectorImplementation(parser,
configuration);
        accessibleSelectors.fixSelectors();

        var accessibleTables = new
hatemile.implementation.AccessibleTableImplementation(parser,
configuration);
        accessibleTables.fixAssociationCellsTable();
    </script>
</body>
</html>

```

Saída em texto para o navegador Mozilla Firefox:

```

HaTeMiLe Tests - Mozilla Firefox
HaTeMiLe Tests
lista com 4 itens
•
linque Alt+Shift+3
Form 1
•
linque
Form 2
•
linque Alt+Shift+9
Shortcuts
•
linque Alt+Shift+0
Summary
fora de lista
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert

botão
Alert

```

botão  
Alert

clicável  
Alert

clicável  
Alert  
clicável  
Alert  
clicável  
Console  
separador  
linque  
Console

botão  
Console

botão  
Console

Console

Console  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1

edição possui autocompletar submenu exigido entrada inválida

Field2  
edição entrada inválida exigido  
Field3

edição entrada inválida exigido multilinha  
fora de edição

Field4  
edição entrada inválida exigido multilinha  
fora de edição

Field5

caixa de combinação recolhido entrada inválida exigido  
0

Field6

caixa de combinação recolhido entrada inválida exigido  
0

Field7

botão de rotação  
edição  
0

botão  
Submit  
edição entrada inválida exigido  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Acessibility  
linque  
Acessibility [d]

gráfico Tem descrição longa  
HTML5  
linque  
HTML5 [d]

cabeçalho nível 2  
Test Shortcuts  
linque Alt+Shift+q  
HTML5

linque Alt+Shift+w  
Go to Wikimidia

Field1

edição possui autocompletar Alt+Shift+e

edição possui autocompletar Alt+Shift+r

botão Alt+Shift+t  
Octobiwan

botão Alt+Shift+y  
Reset button

botão Alt+Shift+u  
Show shortcuts

botão Alt+Shift+i  
Subit Button

cabeçalho nível 2  
Test Tables

tabela com 7 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

coluna 4

4

2 linha 2 coluna 3

1

linha 3 coluna 2

1

1 coluna 3

2

1 coluna 4

3

linha 4 1 coluna 1

1 coluna 2

1 coluna 3

1 coluna 4

linha 5 1 coluna 1

1 coluna 2

1 coluna 3

1 coluna 4

linha 6 1 coluna 2

1 coluna 4

linha 7 1 coluna 1

1 coluna 2

1 coluna 3

fora de tabela

tabela com 3 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

linha 2 coluna 1

1

coluna 2

2

3 coluna 4

3

linha 3 coluna 1

1

coluna 2

2

3 coluna 3

3

3 coluna 4

4

fora de tabela

Summary:

lista com 1 itens

1.

linque

HaTeMiLe Tests

lista com 5 itens

1.

linque

Test Events

2.

linque  
 Test Forms  
 3.  
 linque  
 Test Images  
 4.  
 linque  
 Test Shortcuts  
 5.  
 linque  
 Test Tables  
 fora de lista  
 fora de lista  
 Shortcuts:  
 lista com 11 itens

- ALT + SHIFT + Q: Go to HTML5
- ALT + SHIFT + W: Go to Wikimidia
- ALT + SHIFT + E: Field1
- ALT + SHIFT + R: Field 2
- ALT + SHIFT + T: Octobiwan
- ALT + SHIFT + Y: Reset button
- ALT + SHIFT + U: Show shortcuts
- ALT + SHIFT + I: Subit Button
- ALT + SHIFT + 3: Form 1
- ALT + SHIFT + 9: Shortcuts
- ALT + SHIFT + 0: Summary

Saída em texto para o navegador Google Chrome:

HaTeMiLe Tests documento  
 HaTeMiLe Tests  
 cabeçalho nível 1  
 HaTeMiLe Tests  
 cabeçalho nível 2  
 Test Events  
 linque  
 Alert  
 botão  
 Alert  
 botão  
 Alert  
 AlertAlert  
 Alert  
 Console



linque  
 Console  
 botão  
 Console  
 botão  
 Console  
 ConsoleConsole  
 Console  
 Drag-and-Drop  
 cabeçalho nível 2  
 Test Forms  
 Field1  
 edição entrada inválida exigido

Field2  
 edição entrada inválida exigido  
 Field3  
 ield1  
 edição entrada inválida exigido

Field2  
 edição entrada inválida exigido  
 Field3  
 edição entrada inválida exigido multilinha  
 fora de edição  
 Field4  
 edição entrada inválida exigido multilinha  
 fora de edição

Field5  
 caixa de combinação recolhido entrada inválida exigido  
 0

Field6  
 caixa de combinação recolhido entrada inválida exigido  
 0

Field7  
 botão de rotação editável  
 0  
 botão de rotação editável  
 botão de rotação editável  
 botão editável

botão editável

botão  
 Submit  
 edição entrada inválida exigido

cabeçalho nível 2

Test Images

gráfico

Acessibility

linque

Acessibility [d]

gráfico

HTML5

linque

HTML5 [d]

cabeçalho nível 2

Test Shortcuts

linque Alt+q

HTML5

linque Alt+w

Go to Wikimidia

Field1

edição Alt+e

edição Alt+r

botão Alt+t

Octobiwan

botão Alt+y

Reset button

botão Alt+u

Show shortcuts

botão Alt+i

Subit Button

cabeçalho nível 2

Test Tables

tabela com 7 linhas e 4 colunas

linha 1 1

1  
1 coluna 1  
1  
2  
2  
1 coluna 2  
2  
3  
1  
2 coluna 3  
3  
1 coluna 4  
4  
linha 2 3  
1  
2 coluna 3  
1  
linha 3 2  
2  
1 coluna 2  
1  
3  
1  
2 coluna 3  
2  
1 coluna 4  
3  
linha 4 1  
1  
1 coluna 1  
  
2  
2  
1 coluna 2  
  
3  
1  
2 coluna 3  
  
1 coluna 4  
  
linha 5 1  
1  
1 coluna 1  
  
2

2  
1 coluna 2

3  
1  
2 coluna 3

1 coluna 4

linha 6 2  
2  
1 coluna 2  
1 coluna 4

linha 7 1  
1  
1 coluna 1

2  
2  
1 coluna 2

3  
1  
2 coluna 3

fora de tabela

1

2

3

1

2

3

1

2

3

4

Summary:

lista com 1 itens

1

linque

HaTeMiLe Tests

lista com 5 itens

1

linque

Test Events

2

linque

Test Forms

3

linque

Test Images

4

linque

Test Shortcuts

5

linque

Test Tables

fora de lista

fora de lista

Shortcuts:

lista com 11 itens

•

ALT + Q: Go to HTML5

•

ALT + W: Go to Wikimidia

•

ALT + E: Field1

•

ALT + R: Field 2

•

ALT + T: Octobiwan

•

ALT + Y: Reset button

•

ALT + U: Show shortcuts

•

ALT + I: Subit Button

•

ALT + 3: Form 1

•

ALT + 9: Shortcuts

•

ALT + 0: Summary

Saída em texto para o navegador Opera:

HaTeMiLe Tests documento  
HaTeMiLe Tests

lista com 4 itens

- 

linque Alt+3

Form 1

- 

linque

Form 2

- 

linque Alt+9

Shortcuts

- 

linque Alt+0

Summary

fora de lista

cabeçalho nível 1

HaTeMiLe Tests

cabeçalho nível 2

Test Events

linque

Alert

botão

Alert

botão

Alert

Alert

Alert

Alert

Console

linque

Console

botão

Console

botão

Console

Console

Console

Console

Drag-and-Drop

cabeçalho nível 2

Test Forms

Field1

edição entrada inválida exigido

Field2

edição entrada inválida exigido

Field3

edição entrada inválida exigido multilinha  
fora de edição

Field4

edição entrada inválida exigido multilinha  
fora de edição

Field5

caixa de combinação recolhido entrada inválida exigido  
0

Field6

caixa de combinação recolhido entrada inválida exigido  
0

Field7

botão de rotação editável

botão de rotação editável

botão editável

botão editável

botão

Submit

edição entrada inválida exigido  
cabeçalho nível 2

Test Images

linque

Acessibility [d]

linque

HTML5 [d]

cabeçalho nível 2

Test Shortcuts

linque Alt+q

HTML5

linque Alt+w

Go to Wikimidia

Field1

edição Alt+e

edição Alt+r

botão Alt+y

Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 4 colunas

linha 1 1

1

1 coluna 1

1

2

2

1 coluna 2

2

3

1

2 coluna 3

3

1 coluna 4

4

linha 2 3

1

2 coluna 3

1

linha 3 2

2

1 coluna 2

1

3

1

2 coluna 3

2

1 coluna 4

3

linha 4 1

1

1 coluna 1

2

2



1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 5 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 6 2

2

1 coluna 2

1 coluna 4

linha 7 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

fora de tabela

1

2

3

1

2

3

1

2

3

4

Summary:

lista com 1 itens

1

linque

HaTeMiLe Tests

lista com 5 itens

1

linque

Test Events

2

linque

Test Forms

3

linque

Test Images

4

linque

Test Shortcuts

5

linque

Test Tables

fora de lista

fora de lista

Shortcuts:

lista com 11 itens

- 

ALT + Q: Go to HTML5

- 

ALT + W: Go to Wikimidia

- 

ALT + E: Field1

- 

ALT + R: Field 2

- 

ALT + T: Octobiwan

- 

ALT + Y: Reset button

- 

ALT + U: Show shortcuts

- 

ALT + I: Subit Button

- 
- ALT + 3: Form 1
- 
- ALT + 9: Shortcuts
- 
- ALT + 0: Summary

Saída em texto para o navegador Internet Explorer:

```

HaTeMiLe Tests - Internet Explorer
HaTeMiLe Tests documento
HaTeMiLe Tests
lista com 4 itens
•
linque Alt+3
Form 1
•
linque
Form 2
•
linque Alt+9
Shortcuts
•
linque Alt+0
Summary
fora de lista
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert
botão
Alert
botão
Alert
clícável
Alert
edição clícável
Alert
clícável
Alert
clícável
Console

```

separador  
linque  
Console  
botão  
Console  
botão  
Console  
Console  
edição  
Console  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1  
edição exigido  
  
Field2  
edição exigido  
Field3  
edição multilinha exigido  
fora de edição  
Field4  
edição multilinha exigido  
fora de edição  
Field5  
caixa de combinação recolhido exigido  
0  
Field6  
caixa de combinação recolhido exigido  
0  
Field7  
botão de rotação editável  
  
botão  
Submit  
edição exigido  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Acessibility  
linque  
Acessibility [d]  
gráfico Tem descrição longa  
HTML5  
linque

HTML5 [d]  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+q  
HTML5

linque Alt+w  
Go to Wikimidia

Field1  
edição Alt+e

edição Alt+r  
botão Alt+t  
Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 3 colunas  
linha 1 coluna 1  
1  
coluna 2  
2  
coluna 3  
3  
coluna 4  
4  
1 linha 2 3 coluna 3  
1

1 linha 3 2 coluna 2

1

3 coluna 3

2

coluna 4

3

linha 4 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 5 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 6 2 coluna 2

coluna 4

linha 7 1 coluna 1

2 coluna 2

3 coluna 3

fora de tabela

tabela com 3 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

linha 2 1 coluna 1

1

coluna 2

2

3 coluna 4

3

linha 3 1 coluna 1

1

coluna 2

2

3 coluna 3

3

3 coluna 4

4

fora de tabela

Summary:

lista com 1 itens

1.

linque

HaTeMiLe Tests

lista com 5 itens

1.

linque

Test Events

2.

linque

Test Forms

3.

linque

Test Images

4.

linque

Test Shortcuts

5.

linque

Test Tables

fora de lista

fora de lista

Shortcuts:

lista com 11 itens

- ALT + Q: Go to HTML5
- ALT + W: Go to Wikimidia
- ALT + E: Field1
- ALT + R: Field 2
- ALT + T: Octobiwan
- ALT + Y: Reset button
- ALT + U: Show shortcuts
- ALT + I: Subit Button
- ALT + 3: Form 1
- ALT + 9: Shortcuts
- ALT + 0: Summary

## Página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe que utiliza o CSS

Código-fonte da página que testa todas as funcionalidades do HaTeMiLe, que utiliza o HaTeMiLe que utiliza o CSS:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HaTeMiLe Tests</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" type="text/css"
href="css/common_elements.css" />
  </head>
  <body>
    <h1>HaTeMiLe Tests</h1>
    <!-- Events -->
    <div>
      <h2>Test Events</h2>
      <a href="#" onclick="alert('Alert A')">Alert</a>
      <button onclick="alert('Alert Button')">Alert</button>
      <input type="button" onclick="alert('Alert Input')"
value="Alert" />
      <span onclick="alert('Alert Span')" style="background:
red;">Alert</span>
      <i onclick="alert('Alert I')">Alert</i>
      <div style="height: 300px; width: 300px; border: 1px solid black"
onclick="alert('Alert Div')">
        Alert
      </div>
      <span onclick="alert('Alert span')"
onkeypress="console.log('Console SPAN')" style="background:
blueviolet;">Console</span>
      <hr />
      <a href="#" onmouseover="console.log('Over A')"
onmouseout="console.log('Out A')">Console</a>
      <button onmouseover="console.log('Over Button')"
onmouseout="console.log('Out Button')">Console</button>
      <input type="button" onmouseover="console.log('Over Input')"
value="Console" onmouseout="console.log('Out Input')" />
      <span onmouseover="console.log('Over Span')"
style="background: red;" onmouseout="console.log('Out
Span')">Console</span>
      <i onmouseover="console.log('Over I')"
```



```

onmouseout="console.log('Out I!')">Console</i>
    <div style="height: 300px; width: 300px; border: 1px solid black"
ondrop="event.preventDefault();event.target.appendChild(document.getE
lementById(event.dataTransfer.getData('text')));"
ondragover="event.preventDefault();">
    Console
    </div>
    <span id="draggable-item"
ondragstart="event.dataTransfer.setData('text', event.target.id);"
draggable="true">Drag-and-Drop</span>
    </div>
    <!-- Forms -->
    <form autocomplete="off" id="form1">
        <h2>Test Forms</h2>
        <label for="field1">Field1</label>
        <input type="text" value="" required="required" id="field1"
autocomplete="on" />
        <label>
            Field2
        <div>
            <input type="text" value="" required="required"
autocomplete="off" />
        </div>
        </label>
        <label for="field3">Field3</label>
        <textarea required="required" id="field3"
autocomplete></textarea>
        <label>
            Field4
            <textarea required="required"
autocomplete="none"></textarea>
        </label>
        <label for="field5">Field5</label>
        <select required="required" id="field5">
            <option value="">0</option>
            <option value="1">1</option>
            <option value="2">2</option>
        </select>
        <label>
            Field6
            <select required="required">
                <option value="">0</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>
        </label>

```

```

        <label for="field7">Field7</label>
        <input type="number" min="0" value="0" max="10"
id="field7" />
        <input type="submit" value="Submit" />
    </form>
    <input type="text" value="" required="" form="form1" />
    <!-- Images -->
    <div>
        <h2>Test Images</h2>
        
        
    </div>
    <!-- Shortcuts -->
    <form action="http://www.webplatform.org/">
        <h2>Test Shortcuts</h2>
        <a href="http://www.w3.org/html/" title="Go to HTML5"
accesskey="q">HTML5</a><br />
        <a href="https://www.wikimedia.org/" accesskey="w">Go to
Wikimidia</a><br />
        <label id="label1">Field1</label>
        <input type="text" value="" aria-labelledby="label1"
accesskey="e" /><br />
        <input type="text" value="" aria-label="Field 2" accesskey="r"
/><br />
        <input type="image"
src="https://octodex.github.com/images/octobiwan.jpg" alt="Octobiwan"
accesskey="t" /><br />
        <input type="reset" value="Reset button" accesskey="y"
/><br />
        <input type="button" value="Show shortcuts" accesskey="u"
onclick="alert('Only in client-side version.');" /><br />
        <input type="submit" value="Subit Button" accesskey="i"
/><br />
    </form>
    <!-- Selectors -->

    <!-- Tables -->
    <div>
        <h2>Test Tables</h2>

```

```

<table>
  <thead>
    <tr>
      <th rowspan="3">1</th>
      <th rowspan="2">2</th>
      <th>3</th>
      <td>4</td>
    </tr>
    <tr>
      <th colspan="2">1</th>
    </tr>
    <tr>
      <th>1</th>
      <th>2</th>
      <td>3</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td></td>
      <td></td>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td rowspan="2"></td>
      <td></td>
      <th></th>
      <td></td>
    </tr>
    <tr>
      <td colspan="2"></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
      <td></td>
    </tr>
  </tbody>
</table>
<table>
  <tr>
    <th>1</th>
    <td>2</td>
    <th colspan="2">3</th>
  </tr>

```

```

        </tr>
        <tr>
            <td>1</td>
            <td colspan="2">2</td>
            <td>3</td>
        </tr>
        <tr>
            <td>1</td>
            <th>2</th>
            <td>3</td>
            <td>4</td>
        </tr>
    </table>
</div>
    <script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/common.js"></script>
    <script type="text/javascript" src="js/config.js"></script>
    <script type="text/javascript"
src="js/eventlistener.js"></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleEventImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleFormImplementation.js"></sc
ript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleImageImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleNavigationImplementation.js"
></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleSelectorImplementation.js"><
/script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleTableImplementation.js"></sc
ript>
    <script type="text/javascript"
src="js/hatemile/util/CommonFunctions.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/Configure.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/SelectorChange.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/Skipper.js"></script>
    <script type="text/javascript"

```

```

src="js/hatemile/util/jqueryandvanilla/JQueryHTMLDOMParser.js"></script>
<script type="text/javascript"
src="js/hatemile/util/jqueryandvanilla/VanillaHTMLDOMElement.js"></script>
<script type="text/javascript">
    var configuration = new hatemile.util.Configure(config);
    var parser = new
hatemile.util.jqueryandvanilla.JQueryHTMLDOMParser(document);

    var accessibleEvent = new
hatemile.implementation.AccessibleEventImplementation(parser,
configuration);
    accessibleEvent.fixDragsandDrops();
    accessibleEvent.fixHovers();
    accessibleEvent.fixActives();

    var accessibleForm = new
hatemile.implementation.AccessibleFormImplementation(parser,
configuration);
    accessibleForm.fixRequiredFields();
    accessibleForm.fixRangeFields();
    accessibleForm.fixAutoCompleteFields();
    accessibleForm.fixLabels();

    var accessibleImage = new
hatemile.implementation.AccessibleImageImplementation(parser,
configuration);
    accessibleImage.fixLongDescriptions();

    var accessibleNavigation = new
hatemile.implementation.AccessibleNavigationImplementation(parser,
configuration, navigator.userAgent);
    accessibleNavigation.fixHeadings();
    accessibleNavigation.fixShortcuts();
    accessibleNavigation.fixSkippers();

    var accessibleSelectors = new
hatemile.implementation.AccessibleSelectorImplementation(parser,
configuration);
    accessibleSelectors.fixSelectors();

    var accessibleTables = new
hatemile.implementation.AccessibleTableImplementation(parser,
configuration);
    accessibleTables.fixAssociationCellsTable();

```

```
        </script>
    </body>
</html>
```

Saída em texto para o navegador Mozilla Firefox:

```
HaTeMiLe Tests - Mozilla Firefox
HaTeMiLe Tests
lista com 4 itens
•
linque Alt+Shift+3
Form 1
•
linque
Form 2
•
linque Alt+Shift+9
Shortcuts
•
linque Alt+Shift+0
Summary
fora de lista
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert

botão
Alert

botão
Alert

clícável
Alert

clícável
Alert
clícável
Alert
clícável
Console
```

separador  
linque  
Console

botão  
Console

botão  
Console

Console

Console  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms

\* Field1 (Autocomplete: List or Inline)

edição possui autocompletar submenu exigido entrada inválida

\* Field2  
edição entrada inválida exigido  
(Autocomplete: None)

\* Field3

edição entrada inválida exigido multilinha  
fora de edição

\* Field4  
edição entrada inválida exigido multilinha  
fora de edição

\* Field5

caixa de combinação recolhido entrada inválida exigido  
0

\* Field6  
caixa de combinação recolhido entrada inválida exigido  
0

Field7 (Min: 0) (Max: 10) (Autocomplete: None)

botão de rotação  
edição  
0

botão  
Submit  
edição entrada inválida exigido  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Acessibility  
linque  
Acessibility [d]

gráfico Tem descrição longa  
HTML5  
linque  
HTML5 [d]

cabeçalho nível 2  
Test Shortcuts  
linque Alt+Shift+q  
HTML5

linque Alt+Shift+w  
Go to Wikimidia

Field1

edição possui autocompletar Alt+Shift+e

edição possui autocompletar Alt+Shift+r  
botão Alt+Shift+t  
Octobiwan

botão Alt+Shift+y  
Reset button

botão Alt+Shift+u  
Show shortcuts



botão Alt+Shift+i  
Subit Button  
cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 4 colunas  
linha 1 coluna 1  
1  
coluna 2  
2  
coluna 3  
3  
coluna 4  
4  
2 linha 2 coluna 3  
1  
linha 3 coluna 2  
1  
1 coluna 3  
2  
1 coluna 4  
3  
linha 4 1 coluna 1  
  
1 coluna 2  
  
1 coluna 3  
  
1 coluna 4  
  
linha 5 1 coluna 1  
  
1 coluna 2  
  
1 coluna 3  
  
1 coluna 4  
  
linha 6 1 coluna 2  
  
1 coluna 4  
  
linha 7 1 coluna 1  
  
1 coluna 2

1 coluna 3

fora de tabela

tabela com 3 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

linha 2 coluna 1

1

coluna 2

2

3 coluna 4

3

linha 3 coluna 1

1

coluna 2

2

3 coluna 3

3

3 coluna 4

4

fora de tabela

Summary:

lista com 1 itens

1.

linque

HaTeMiLe Tests

lista com 5 itens

1.

linque

Test Events

2.

linque

Test Forms

3.

linque

Test Images

4.

linque

Test Shortcuts

5.

linque

Test Tables

fora de lista

fora de lista

Shortcuts:

lista com 11 itens

- ALT + SHIFT + Q: Go to HTML5
- ALT + SHIFT + W: Go to Wikimidia
- ALT + SHIFT + E: Field1
- ALT + SHIFT + R: Field 2
- ALT + SHIFT + T: Octobiwan
- ALT + SHIFT + Y: Reset button
- ALT + SHIFT + U: Show shortcuts
- ALT + SHIFT + I: Subit Button
- ALT + SHIFT + 3: Form 1
- ALT + SHIFT + 9: Shortcuts
- ALT + SHIFT + 0: Summary

Saída em texto para o navegador Google Chrome:

HaTeMiLe Tests documento

HaTeMiLe Tests

lista com 4 itens

•

linque Alt+3

Form 1

•

linque

Form 2

•

linque Alt+9

Shortcuts

•

linque Alt+0

Summary

fora de lista

cabeçalho nível 1

HaTeMiLe Tests

cabeçalho nível 2

Test Events

linque

Alert

botão

Alert

botão

Alert  
 Alert  
 Alert  
 Alert  
 Console  
 linque  
 Console  
 botão  
 Console  
 botão  
 Console  
 Console  
 Console  
 Console  
 Drag-and-Drop  
 cabeçalho nível 2  
 Test Forms  
 \* Field1 (Autocomplete: List or Inline)  
 edição entrada inválida exigido  
  
 \* Field2  
 edição entrada inválida exigido  
 (Autocomplete: None)  
 \* Field3  
 edição entrada inválida exigido multilinha  
 fora de edição  
 \* Field4  
 edição entrada inválida exigido multilinha  
 fora de edição  
 \* Field5  
 caixa de combinação recolhido entrada inválida exigido  
 0  
 \* Field6  
 caixa de combinação recolhido entrada inválida exigido  
 0  
 Field7 (Min: 0) (Max: 10) (Autocomplete: None)  
 botão de rotação editável  
 botão de rotação editável  
 botão editável  
  
 botão editável  
  
 botão  
 Submit  
 edição entrada inválida exigido  
 cabeçalho nível 2

Test Images

linque

Acessibility [d]

linque

HTML5 [d]

cabeçalho nível 2

Test Shortcuts

linque Alt+q

HTML5

linque Alt+w

Go to Wikimidia

Field1

edição Alt+e

edição Alt+r

botão Alt+t

Octobiwan

botão Alt+y

Reset button

botão Alt+u

Show shortcuts

botão Alt+i

Subit Button

cabeçalho nível 2

Test Tables

tabela com 7 linhas e 4 colunas

linha 1 1

1

1 coluna 1

1

2

2

1 coluna 2

2

3

1

2 coluna 3

3

1 coluna 4

4

linha 2 3

1

2 coluna 3

1

linha 3 2

2

1 coluna 2

1

3

1

2 coluna 3

2

1 coluna 4

3

linha 4 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 5 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 6 2

2

1 coluna 2

1 coluna 4

linha 7 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

fora de tabela

1

2

3

1

2

3

1

2

3

4

Summary:

lista com 1 itens

1

linque

HaTeMiLe Tests

lista com 5 itens

1

linque

Test Events

2

linque

Test Forms

3

linque  
 Test Images  
 4  
 linque  
 Test Shortcuts  
 5  
 linque  
 Test Tables  
 fora de lista  
 fora de lista  
 Shortcuts:  
 lista com 11 itens
 

- 
- ALT + Q: Go to HTML5
- 
- ALT + W: Go to Wikimidia
- 
- ALT + E: Field1
- 
- ALT + R: Field 2
- 
- ALT + T: Octobiwan
- 
- ALT + Y: Reset button
- 
- ALT + U: Show shortcuts
- 
- ALT + I: Subit Button
- 
- ALT + 3: Form 1
- 
- ALT + 9: Shortcuts
- 
- ALT + 0: Summary

Saída em texto para o navegador Opera:

HaTeMiLe Tests documento  
 HaTeMiLe Tests  
 lista com 4 itens
 

- 
- linque Alt+3
- Form 1
-



linque  
 Form 2  
 •  
 linque Alt+9  
 Shortcuts  
 •  
 linque Alt+0  
 Summary  
 fora de lista  
 cabeçalho nível 1  
 HaTeMiLe Tests  
 cabeçalho nível 2  
 Test Events  
 linque  
 Alert  
 botão  
 Alert  
 botão  
 Alert  
 Alert  
 Alert  
 Alert  
 Console  
 linque  
 Console  
 botão  
 Console  
 botão  
 Console  
 Console  
 Console  
 Console  
 Console  
 Drag-and-Drop  
 cabeçalho nível 2  
 Test Forms  
 \* Field1 (Autocomplete: List or Inline)  
 edição entrada inválida exigido  
  
 \* Field2  
 edição entrada inválida exigido  
 (Autocomplete: None)  
 \* Field3  
 edição entrada inválida exigido multilinha  
 fora de edição  
 \* Field4  
 edição entrada inválida exigido multilinha

fora de edição

\* Field5

caixa de combinação recolhido entrada inválida exigido  
0

\* Field6

caixa de combinação recolhido entrada inválida exigido  
0

Field7 (Min: 0) (Max: 10) (Autocomplete: None)

botão de rotação editável

botão de rotação editável

botão editável

botão editável

botão

Submit

edição entrada inválida exigido

cabeçalho nível 2

Test Images

gráfico

Acessibility

linque

Acessibility [d]

gráfico

HTML5

linque

HTML5 [d]

cabeçalho nível 2

Test Shortcuts

linque Alt+q

HTML5

linque Alt+w

Go to Wikimidia

Field1

edição Alt+e

edição Alt+r

botão Alt+t

Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 4 colunas  
linha 1 1  
1  
1 coluna 1  
1  
2  
2  
1 coluna 2  
2  
3  
1  
2 coluna 3  
3  
1 coluna 4  
4  
linha 2 3  
1  
2 coluna 3  
1  
linha 3 2  
2  
1 coluna 2  
1  
3  
1  
2 coluna 3  
2  
1 coluna 4  
3  
linha 4 1  
1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 5 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

1 coluna 4

linha 6 2

2

1 coluna 2

1 coluna 4

linha 7 1

1

1 coluna 1

2

2

1 coluna 2

3

1

2 coluna 3

fora de tabela

1

2

3

1

2

3

1

2

3

4

Summary:

lista com 1 itens

1

linque

HaTeMiLe Tests

lista com 5 itens

1

linque

Test Events

2

linque

Test Forms

3

linque

Test Images

4

linque

Test Shortcuts

5

linque

Test Tables

fora de lista

fora de lista

Shortcuts:

lista com 11 itens

•

ALT + Q: Go to HTML5

•

ALT + W: Go to Wikimidia

•

ALT + E: Field1

•

ALT + R: Field 2

•

ALT + T: Octobiwan

•

ALT + Y: Reset button

- 
- ALT + U: Show shortcuts
- 
- ALT + I: Submit Button
- 
- ALT + 3: Form 1
- 
- ALT + 9: Shortcuts
- 
- ALT + 0: Summary

Saída em texto para o navegador Internet Explorer:

```

HaTeMiLe Tests - Internet Explorer
HaTeMiLe Tests documento
HaTeMiLe Tests
lista com 4 itens
•
linque Alt+3
Form 1
•
linque
Form 2
•
linque Alt+9
Shortcuts
•
linque Alt+0
Summary
fora de lista
cabeçalho nível 1
HaTeMiLe Tests
cabeçalho nível 2
Test Events
linque
Alert
botão
Alert
botão
Alert
clícável
Alert
edição clícável
Alert

```

clicável  
Alert  
clicável  
Console  
separador  
linque  
Console  
botão  
Console  
botão  
Console  
Console  
edição  
Console  
Console  
Drag-and-Drop  
cabeçalho nível 2  
Test Forms  
Field1  
edição exigido  
  
Field2  
edição exigido  
Field3  
edição multilinha exigido  
fora de edição  
Field4  
edição multilinha exigido  
fora de edição  
Field5  
caixa de combinação recolhido exigido  
0  
Field6  
caixa de combinação recolhido exigido  
0  
Field7  
botão de rotação editável  
  
botão  
Submit  
edição exigido  
cabeçalho nível 2  
Test Images  
gráfico Tem descrição longa  
Accessibility  
linque

Acessibility [d]  
gráfico Tem descrição longa  
HTML5  
linque  
HTML5 [d]  
cabeçalho nível 2  
Test Shortcuts  
linque Alt+q  
HTML5

linque Alt+w  
Go to Wikimidia

Field1  
edição Alt+e

edição Alt+r  
botão Alt+t  
Octobiwan

botão Alt+y  
Reset button

botão Alt+u  
Show shortcuts

botão Alt+i  
Subit Button

cabeçalho nível 2  
Test Tables  
tabela com 7 linhas e 3 colunas  
linha 1 coluna 1  
1  
coluna 2  
2  
coluna 3  
3



coluna 4

4

1 linha 2 3 coluna 3

1

1 linha 3 2 coluna 2

1

3 coluna 3

2

coluna 4

3

linha 4 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 5 1 coluna 1

2 coluna 2

3 coluna 3

coluna 4

linha 6 2 coluna 2

coluna 4

linha 7 1 coluna 1

2 coluna 2

3 coluna 3

fora de tabela

tabela com 3 linhas e 4 colunas

linha 1 coluna 1

1

coluna 2

2

coluna 3

3

linha 2 1 coluna 1

1

coluna 2  
 2  
 3 coluna 4  
 3  
 linha 3 1 coluna 1  
 1  
 coluna 2  
 2  
 3 coluna 3  
 3  
 3 coluna 4  
 4  
 fora de tabela  
 Summary:  
 lista com 1 itens  
 1.  
 linque  
 HaTeMiLe Tests  
 lista com 5 itens  
 1.  
 linque  
 Test Events  
 2.  
 linque  
 Test Forms  
 3.  
 linque  
 Test Images  
 4.  
 linque  
 Test Shortcuts  
 5.  
 linque  
 Test Tables  
 fora de lista  
 fora de lista  
 Shortcuts:  
 lista com 11 itens  
 • ALT + Q: Go to HTML5  
 • ALT + W: Go to Wikimidia  
 • ALT + E: Field1  
 • ALT + R: Field 2  
 • ALT + T: Octobiwan  
 • ALT + Y: Reset button  
 • ALT + U: Show shortcuts  
 • ALT + I: Subit Button

- ALT + 3: Form 1
- ALT + 9: Shortcuts
- ALT + 0: Summary

### **Página que testa a semântica dos navegadores, que não utiliza o HaTeMiLe**

Código-fonte da página que testa a semântica dos navegadores, porém não utiliza o HaTeMiLe:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HaTeMiLe Tests</title>
    <meta charset="UTF-8" />
  </head>
  <body>
<header>
  <h1>Wake up sheeple!</h1>
  <p><a href="news.html">News</a> -
    <a href="blog.html">Blog</a> -
    <a href="forums.html">Forums</a></p>
  <p>Last Modified: <span itemprop="dateModified">2009-04-
01</span></p>
  <nav>
    <h1>Navigation</h1>
    <ul>
      <li><a href="articles.html">Index of all articles</a></li>
      <li><a href="today.html">Things sheeple need to wake up for
today</a></li>
      <li><a href="successes.html">Sheeple we have managed to
wake</a></li>
    </ul>
  </nav>
</header>
<main>
  <article itemprop="blogPosts" itemscope
itemtype="http://schema.org/BlogPosting">
    <header>
      <h1 itemprop="headline">My Day at the Beach</h1>
    </header>
    <div itemprop="articleBody">
      <p>Today I went to the beach and had a lot of fun.</p>
      ...more content...
    </div>
```

```

<footer>
  <p>Posted <time itemprop="datePublished" datetime="2009-10-
10">Thursday</time>.</p>
</footer>
</article>
...more blog posts...
</main>
<footer>
  <p>Copyright ©
  <span itemprop="copyrightYear">2010</span>
  <span itemprop="copyrightHolder">The Example Company</span>
</p>
  <p><a href="about.html">About</a> -
    <a href="policy.html">Privacy Policy</a> -
    <a href="contact.html">Contact Us</a></p>
</footer>
</body>
</html>

```

Saída em texto para o navegador Mozilla Firefox:

```

HaTeMiLe Tests - Mozilla Firefox
HaTeMiLe Tests
marca bâner
cabeçalho nível 1
Wake up sheeple!
linque
News
-
linque
Blog
-
linque
Forums
Last Modified: 2009-04-01
marca navegação
cabeçalho nível 1
Navigation
lista com 3 itens
•
linque
Index of all articles
•
linque

```

Things sheeple need to wake up for today

- 

linque

Sheeple we have managed to wake

fora de lista

marca principal

cabeçalho nível 1

My Day at the Beach

Today I went to the beach and had a lot of fun.

...more content...

Posted

Thursday

.

...more blog posts...

marca informação de conteúdo

Copyright © 2010 The Example Company

linque

About

-

linque

Privacy Policy

-

linque

Contact Us

Saída em texto para o navegador Google Chrome:

HaTeMiLe Tests

marca bânér

cabeçalho nível 1

Wake up sheeple!

linque

News

-

linque

Blog

-

linque

Forums

Last Modified: 2009-04-01

marca navegação

cabeçalho nível 1

Navigation

lista com 3 itens

- linque  
Index of all articles
- linque  
Things sheeple need to wake up for today
- linque  
Sheeple we have managed to wake  
fora de lista  
marca principal  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted  
Thursday
- ...more blog posts...  
marca informação de conteúdo  
Copyright © 2010The Example Company  
linque  
About  
-  
linque  
Privacy Policy  
-  
linque  
Contact Us

Saída em texto para o navegador Opera:

HaTeMiLe Tests documento  
HaTeMiLe Tests  
marca bâner  
cabeçalho nível 1  
Wake up sheeple!  
linque  
News  
-  
linque  
Blog  
-  
linque

Forums  
Last Modified: 2009-04-01  
marca navegação  
cabeçalho nível 1  
Navigation  
lista com 3 itens  
•  
linque  
Index of all articles  
•  
linque  
Things sheeple need to wake up for today  
•  
linque  
Sheeple we have managed to wake  
fora de lista  
marca principal  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted  
Thursday  
.  
...more blog posts...  
marca informação de conteúdo  
Copyright © 2010The Example Company  
linque  
About  
-  
linque  
Privacy Policy  
-  
linque  
Contact Us

Saída em texto para o navegador Internet Explorer:

HaTeMiLe Tests - Internet Explorer  
HaTeMiLe Tests  
cabeçalho nível 1  
Wake up sheeple!  
linque  
News

-  
linque  
Blog  
-  
linque  
Forums  
Last Modified:  
2009-04-01  
cabeçalho nível 1  
Navigation  
lista com 3 itens  
•  
linque  
Index of all articles  
•  
linque  
Things sheeple need to wake up for today  
•  
linque  
Sheeple we have managed to wake  
fora de lista  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted  
Thursday  
.  
...more blog posts...  
Copyright ©  
2010  
The Example Company  
linque  
About  
-  
linque  
Privacy Policy  
-  
linque  
Contact Us  
My Day at the Beach



Código-fonte da página que testa as semântica dos navegadores, que utiliza o HaTeMiLe:

```
<!DOCTYPE html>
<html>
  <head>
    <title>HaTeMiLe Tests</title>
    <meta charset="UTF-8" />
  </head>
  <body>
<header>
  <h1>Wake up sheeple!</h1>
  <p><a href="news.html">News</a> -
    <a href="blog.html">Blog</a> -
    <a href="forums.html">Forums</a></p>
  <p>Last Modified: <span itemprop="dateModified">2009-04-
01</span></p>
  <nav>
    <h1>Navigation</h1>
    <ul>
      <li><a href="articles.html">Index of all articles</a></li>
      <li><a href="today.html">Things sheeple need to wake up for
today</a></li>
      <li><a href="successes.html">Sheeple we have managed to
wake</a></li>
    </ul>
  </nav>
</header>
<main>
  <article itemprop="blogPosts" itemscope
itemtype="http://schema.org/BlogPosting">
    <header>
      <h1 itemprop="headline">My Day at the Beach</h1>
    </header>
    <div itemprop="articleBody">
      <p>Today I went to the beach and had a lot of fun.</p>
      ...more content...
    </div>
    <footer>
      <p>Posted <time itemprop="datePublished" datetime="2009-10-
10">Thursday</time>.</p>
    </footer>
  </article>
  ...more blog posts...
</main>
<footer>
  <p>Copyright ©
```

```

    <span itemprop="copyrightYear">2010</span>
    <span itemprop="copyrightHolder">The Example Company</span>
  </p>
  <p> <a href="about.html">About</a> -
    <a href="policy.html">Privacy Policy</a> -
    <a href="contact.html">Contact Us</a> </p>
</footer>
<script type="text/javascript" src="js/jquery.js"></script>
    <script type="text/javascript" src="js/common.js"></script>
    <script type="text/javascript" src="js/config.js"></script>
    <script type="text/javascript"
src="js/eventlistener.js"></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleEventImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleFormImplementation.js"></sc
ript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleImageImplementation.js"></s
cript>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleNavigationImplementation.js"
></script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleSelectorImplementation.js"><
/script>
    <script type="text/javascript"
src="js/hatemile/implementation/AccessibleTableImplementation.js"></sc
ript>
    <script type="text/javascript"
src="js/hatemile/util/CommonFunctions.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/Configure.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/SelectorChange.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/Skipper.js"></script>
    <script type="text/javascript"
src="js/hatemile/util/jqueryandvanilla/JQueryHTMLDOMParser.js"></scrip
t>
    <script type="text/javascript"
src="js/hatemile/util/jqueryandvanilla/VanillaHTMLDOMElement.js"></scri
pt>
    <script type="text/javascript">
        window.addEventListener('load', function () {

```

```

        var configuration = new
hatemile.util.Configure(config);
        var parser = new
hatemile.util.jqueryandvanilla.JQueryHTMLDOMParser(document);

        var accessibleEvent = new
hatemile.implementation.AccessibleEventImplementation(parser,
configuration);

        accessibleEvent.fixDragsandDrops();
        accessibleEvent.fixHovers();
        accessibleEvent.fixActives();

        var accessibleForm = new
hatemile.implementation.AccessibleFormImplementation(parser,
configuration);

        accessibleForm.fixRequiredFields();
        accessibleForm.fixRangeFields();
        accessibleForm.fixAutoCompleteFields();
        accessibleForm.fixLabels();

        var accessibleImage = new
hatemile.implementation.AccessibleImageImplementation(parser,
configuration);

        accessibleImage.fixLongDescriptions();

        var accessibleNavigation = new
hatemile.implementation.AccessibleNavigationImplementation(parser,
configuration);

        accessibleNavigation.fixHeadings();
        accessibleNavigation.fixShortcuts();
        accessibleNavigation.fixSkippers();

        var accessibleSelectors = new
hatemile.implementation.AccessibleSelectorImplementation(parser,
configuration);

        accessibleSelectors.fixSelectors();

        var accessibleTables = new
hatemile.implementation.AccessibleTableImplementation(parser,
configuration);

        accessibleTables.fixAssociationCellsTable();
    });
</script>
</body>
</html>

```

Saída em texto para o navegador Mozilla Firefox:

HaTeMiLe Tests - Mozilla Firefox  
HaTeMiLe Tests  
lista com 2 itens  
•  
linque Alt+Shift+1  
Main Content  
•  
linque Alt+Shift+2  
Menu  
fora de lista  
marca bâner  
cabeçalho nível 1  
Wake up sheeple!  
linque  
News  
-  
linque  
Blog  
-  
linque  
Forums  
Last Modified: 2009-04-01  
marca navegação  
cabeçalho nível 1  
Navigation  
lista com 3 itens  
•  
linque  
Index of all articles  
•  
linque  
Things sheeple need to wake up for today  
•  
linque  
Sheeple we have managed to wake  
fora de lista  
marca principal  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted

Thursday

.

...more blog posts...

marca informação de conteúdo

Copyright © 2010 The Example Company

linque

About

-

linque

Privacy Policy

-

linque

Contact Us

Saída em texto para o navegador Google Chrome:

HaTeMiLe Tests  
lista com 2 itens

•

linque Alt+1

Main Content

•

linque Alt+2

Menu

fora de lista

marca bâner

cabeçalho nível 1

Wake up sheeple!

linque

News

-

linque

Blog

-

linque

Forums

Last Modified: 2009-04-01

marca navegação

cabeçalho nível 1

Navigation

lista com 3 itens

•

linque

Index of all articles

- linque  
Things sheeple need to wake up for today
- linque  
Sheeple we have managed to wake  
fora de lista  
marca principal  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted  
Thursday
- ...more blog posts...  
marca informação de conteúdo  
Copyright © 2010The Example Company  
linque  
About  
-  
linque  
Privacy Policy  
-  
linque  
Contact Us

Saída em texto para o navegador Opera:

HaTeMiLe Tests documento  
HaTeMiLe Tests  
lista com 2 itens

- linque Alt+1  
Main Content
- linque Alt+2  
Menu  
fora de lista  
marca banner  
cabeçalho nível 1  
Wake up sheeple!  
linque  
News

-  
linque  
Blog  
-  
linque  
Forums  
Last Modified: 2009-04-01  
marca navegação  
cabeçalho nível 1  
Navigation  
lista com 3 itens  
•  
linque  
Index of all articles  
•  
linque  
Things sheeple need to wake up for today  
•  
linque  
Sheeple we have managed to wake  
fora de lista  
marca principal  
cabeçalho nível 1  
My Day at the Beach  
Today I went to the beach and had a lot of fun.  
...more content...  
Posted  
Thursday  
.  
...more blog posts...  
marca informação de conteúdo  
Copyright © 2010The Example Company  
linque  
About  
-  
linque  
Privacy Policy  
-  
linque  
Contact Us

Saída em texto para o navegador Internet Explorer:

HaTeMiLe Tests  
lista com 2 itens

- 

linque Alt+1  
Main Content

- 

linque Alt+2  
Menu

fora de lista  
cabeçalho nível 1  
Wake up sheeple!

linque  
News

- 

linque  
Blog

- 

linque  
Forums

Last Modified:  
2009-04-01

marca navegação  
cabeçalho nível 1  
Navigation  
lista com 3 itens

- 

linque  
Index of all articles

- 

linque  
Things sheeple need to wake up for today

- 

linque  
Sheeple we have managed to wake  
fora de lista

marca principal  
cabeçalho nível 1  
My Day at the Beach

Today I went to the beach and had a lot of fun.  
...more content...

Posted  
Thursday

- 

...more blog posts...

Copyright ©  
2010



The Example Company

linque

About

-

linque

Privacy Policy

-

linque

Contact Us

My Day at the Beach

## ANEXO I – E-MAILS DA EQUIPE DE REVISORES DAS EXTENSÕES DA MOZILLA

Os e-mails abaixo estão ordenados em ordem cronológica e os nomes dos revisores foram substituídos por #####, pois não foi solicitado aos mesmos permissão para divulgação de seus nomes.

### E-mail informando o motivo da recusa da versão 0.1.4 do HaTeMiLe for Browser

**Título:** *Mozilla Add-ons: HaTeMiLe for Browser 0.1.4 Rejected*

**Enviado em:** 5 de dezembro de 2014 19:48

*Your add-on, HaTeMiLe for Browser 0.1.4, has been reviewed by an editor and did not meet the criteria for being hosted in our gallery.*

*Reviewer:*

#####

*Comments:*

*Your version was rejected because of the following problems:*

*1) Generating script fragments from unsanitized string data is error prone and poses a major risk of security vulnerabilities. For more information, please see [https://developer.mozilla.org/en-US/docs/XUL/School\\_tutorial/DOM\\_Building\\_and\\_HTML\\_Insertion#listeners](https://developer.mozilla.org/en-US/docs/XUL/School_tutorial/DOM_Building_and_HTML_Insertion#listeners)*

*In any case, please do not generate complex script fragments in such a manner at all.*

*2) Your add-on creates DOM nodes from HTML strings containing unsanitized data, by assigning to `innerHTML` or through similar means. Aside from being inefficient, this is a major security risk. For more information, see [https://developer.mozilla.org/en/XUL\\_School/DOM\\_Building\\_and\\_HTML\\_Insertion](https://developer.mozilla.org/en/XUL_School/DOM_Building_and_HTML_Insertion)*

*Please note that, when combined with the above, values need to be escaped for both HTML and JavaScript.*

*Please also assign to `textContent` rather than `innerHTML` where applicable.*

*Please fix them and submit again. Thank you.*

*Tested on Linux with Firefox 30.0*

*This version of your add-on has been disabled. You may re-request review by addressing the editor's comments and uploading a new version. To learn more about the review process, please visit <https://addons.mozilla.org/developers/docs/policies/reviews#selection>*

*If you have any questions or comments on this review, please reply to this email or join #amo-editors on [irc.mozilla.org](https://irc.mozilla.org)*

--

*Mozilla Add-ons*

*<https://addons.mozilla.org>*

=====

*You received this email because you're subscribed to:*

*\* email me when my add-on is reviewed by an editor*

*You can't unsubscribe from this type of email.*

*Manage email notifications: <https://addons.mozilla.org/users/edit#acct-notify>*

**E-mail informando o motivo da versão 1.0.0 do HaTeMiLe for Browser não ter sido aprovada totalmente**

**Título:** Mozilla Add-ons: HaTeMiLe for Browser 1.0.0 Preliminary Reviewed

**Enviado em:** 27 de agosto de 2015 06:36

*Your add-on, HaTeMiLe for Browser 1.0.0, has been reviewed by an editor and did not meet the criteria for full review. However, your add-on has been granted preliminary review and is now available for download in our gallery at <https://addons.mozilla.org/addon/hatemile-for-browser/>*

*Reviewer:*

*#####*

*Comments:*

*This version didn't pass full review because of the following issues:*

*1. Fully reviewed and listed on AMO add-ons have to have description in the Add-on Details section.*

*You need to correct them to get full approval. Thanks.*

*Your add-on will now appear in search results and categories with some limitations. You may re-request full review by addressing the editor's comments and uploading a new version. To learn more about the review process, please visit*  
<https://addons.mozilla.org/developers/docs/policies/reviews#selection>

*If you have any questions or comments on this review, please reply to this email or join #amo-editors on irc.mozilla.org*

--

*Mozilla Add-ons*  
<https://addons.mozilla.org>

=====

*You received this email because you're subscribed to:*  
 \* email me when my add-on is reviewed by an editor

*You can't unsubscribe from this type of email.*

*Manage email notifications:* <https://addons.mozilla.org/users/edit#acct-notify>

**E-mail informando o motivo da versão 1.0.0 do HaTeMiLe for Browser não ter sido aprovada totalmente, após ter sido informado os detalhes da versão**

**Título:** Mozilla Add-ons: HaTeMiLe for Browser 1.0.0 Preliminary Reviewed

**Enviado em:** 21 de setembro de 2015 18:48

*Your add-on, HaTeMiLe for Browser 1.0.0, has been reviewed by an editor and did not meet the criteria for full review. However, your add-on has been granted preliminary review and is now available for download in our gallery at <https://addons.mozilla.org/addon/hatemile-for-browser/>*

*Reviewer:*  
 #####

*Comments:*

*This version didn't pass full review because of the following issues:*

- 1) Your add-on contains duplicate or hidden files or folders (nbproject). These files complicate the review process and may contain sensitive information about the system you created the XPI on. Please correct your packaging process so that these files aren't included.*
- 2) Your add-on claims to be compatible with Firefox 4. However, it uses the Event constructor in content/client\_side/hatemile/implementation/AccessibleEventImplementation.js line 339. The Event constructor is only supported starting with Firefox 11. Side note: you should be using the MouseEvent constructor if the goal is constructing a proper mouse event object (see <https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent/MouseEvent>).*
- 3) In order to prevent conflicts with webpage functionality, you need to wrap your "loose" variables and functions within a JavaScript object. You can see examples on how to do this at [https://developer.mozilla.org/en/XUL\\_School/JavaScript\\_Object\\_Management](https://developer.mozilla.org/en/XUL_School/JavaScript_Object_Management). Your extension currently injects scripts into webpages that will define the following variables outside of namespaces: versionCompare, isEmpty, configure, parser, accessibleEvent, accessibleForm, accessibleImage, accessibleShortcut, accessibleTable, exports, \_base, \_base1*
- 4) While your extension restores the \$ variable if already defined in the page, the jQuery variable will be overwritten and might cause web pages to break. This is particularly problematic given that you use a very outdated jQuery version (1.6.4) which might result in security issues on these websites.*
- 5) The use of jQuery.find() with dynamically generated selectors (e.g. in content/client\_side/hatemile/implementation/AccessibleNavigationImplementation.js line 468) is a potential security issue, especially with jQuery versions before 1.9. Older jQuery versions will interpret any selector with HTML tags inside it as HTML code, this could result in JavaScript execution. I recommend making JQueryHTMLDOMParser use element.querySelectorAll() instead of jQuery.find().*

*You need to correct them to get full approval. Note that issues 3) and 4) can be prevented completely by using a proper content script mechanism. Content scripts run separately from the webpage's JavaScript and can only interact with DOM objects, JavaScript code running on the page stays "invisible". As far as I can tell you are only interacting with the page's JavaScript in order to find event handlers - a better approach here would be using the event listener service (<https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XPCOM/Reference/Interface/nsIEventListenerService>). That will also solve the problem that you are currently not seeing event handlers added while the page is being loaded.*

*You can use frame scripts in order to load your script into all pages ([https://developer.mozilla.org/en-US/Firefox/Multiprocess\\_Firefox/Frame\\_script\\_loading\\_and\\_lifetime](https://developer.mozilla.org/en-US/Firefox/Multiprocess_Firefox/Frame_script_loading_and_lifetime) has an example, you would use the global message manager instead of a message manager for a single tab). You can use data: URIs to combine multiple script files into one:*

`mm.loadFrameScript("data:text/javascript," + encodeURIComponent("script"), false).`

You can also use the Stylesheet Service instead of loading styles into each webpage individually: [https://developer.mozilla.org/en-US/docs/Using\\_the\\_Stylesheet\\_Service](https://developer.mozilla.org/en-US/docs/Using_the_Stylesheet_Service)

Other issues worth noting:

1) Function `addslashes()` in `content/auxiliar_functions.js` is incomplete, `\u2028` and `\u2029` have special meaning in JavaScript strings as well and need to be escaped.

2) You are moving localization strings into preferences on first run (`content/hatemile_browser.js` line 21). This assumes that the browser language (and consequently the locale used by the user) never changes - which is not true. You should just read the strings from the string bundle when you need them. If the goal is allowing to override these strings via preferences then you can leave the preferences empty - they should only override the localization strings if set. On the other hand, you currently have `selectorchanges` and `skippers` variables in locales - these don't depend on the locale and really don't belong there.

3) Function `isEmpty()` in `content/client_side/common.js` will return true for any object as long as it isn't an array - including `Element.prototype.eventListenerList` that it is being used on.

4) Function `enterPressed()` in `content/client_side/hatemile/implementation/AccessibleEventImplementation.js` compares key codes to char codes, that's wrong. While both are identical in many cases, they are really different concepts. You should really be comparing the key code to `KeyEvent.DOM_VK_RETURN`.

5) You are using `.find("body *")` on several occasions to get all elements. The better approach would be using a `TreeWalker` which is more efficient because it won't create a list immediately - see <https://developer.mozilla.org/en-US/docs/Web/API/Document/createTreeWalker>. In fact, in many cases you can completely avoid iterating through all elements - just attach your event handler on `document.body`, most events will bubble up anyway. You can then look at `event.target` to see which element received the event.

Thanks.

Your add-on will now appear in search results and categories with some limitations. You may re-request full review by addressing the editor's comments and uploading a new version. To learn more about the review process, please visit <https://addons.mozilla.org/developers/docs/policies/reviews>

If you have any questions or comments on this review, please reply to this email or join `#amo-editors` on [irc.mozilla.org](https://irc.mozilla.org)

--

*Mozilla Add-ons*  
*<https://addons.mozilla.org>*

=====

*You received this email because you're subscribed to:*  
*\* email me when my add-on is reviewed by an editor*

*You can't unsubscribe from this type of email.*

*Manage email notifications: <https://addons.mozilla.org/users/edit#acct-notify>*